NASA CR- 132698

REACTING FLUIDS LABORATORY

(NASA-CR-132698) FINITE-RATE CHEMISTRY EFFECTS UPON CONVECTIVE AND RADIATIVE HEATING OF AN ATMOSPHERIC ENTRY VEHICLE (Louisiana State Univ.) 336 p HC $9.50 CSCL 22C G3/13
N75-28089
Unclas
31078

FINITE-RATE CHEMISTRY EFFECTS UPON CONVECTIVE AND RADIATIVE HEATING OF AN ATMOSPHERIC ENTRY VEHICLE

by
Guillermo Perez
under the direction of
Richard C. Farmer
&
Ralph W. Pike

Prepared Under
Grant NGR 19-001-059

by
Department of Chemical Engineering
Louisiana State University
Baton Rouge, Louisiana 70803

for

Langley Research Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

LOUISIANA STATE UNIVERSITY

FINITE-RATE CHEMISTRY EFFECTS UPON CONVECTIVE AND RADIATIVE HEATING OF AN ATMOSPHERIC ENTRY VEHICLE

by

Guillermo Perez
under the direction of
Richard C. Farmer
&
Ralph W. Pike

Prepared Under
Grant NGR 19-001-059

by

Department of Chemical Engineering
Louisiana State University
Baton Rouge, Louisiana 70803

for

Langley Research Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

PREFACE

This report gives the results for a finite-rate, stagnation-line analysis of the radiative heating of a phenolic-nylon ablator. The analysis includes flowfield coupling with the ablator surface, binary diffusion, and a coupled line and continuum radiation model. This report serves as a user's manuel and operating instructions for the computer programs listed in this document. This analysis has been incorporated into an around-the-body analysis by these same principal investigators. Copies of the decks which are used to accomplish this later analysis have been supplied to Dr. James N. Moss, grant monitor, of the NASA, Langley Research Center.

This report also served as Guillermo Perez's dissertation requirement in obtaining a Doctor of Philosophy degree in Chemical Engineering.

ACKNOWLEDGEMENTS

The authors express their appreciation to Mr. Robert T. Swann of NASA who was responsible for initiating the study and whose interest and consultation was very valuable as the work progressed. Also the efforts of Dr. James N. Moss, the grant monitor, are gratefully acknowledged. Also we are most appreciative of the helpful discussions with Mr. K. H. Wilson of the Lockheed Missles and Space Company and Dr. W. S. Rigdon and Dr. R. B. Dirling of the McDonnell-Douglas Astronautics Company.

The typing of this manuscript was very capably performed by Madeleine Perez, Joi Holmes, Sandy Prestridge, and Gayle Smith.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

English

B_v	Planckian radiation intensity (m/t^2x no. of particles)$
C_i	Mass fraction (mass of i/unit mass of fluid) $C_i = \frac{\rho i}{\rho}$, $\sum_i Ci = 1$
C_p	Specific heat at constant pressure (L^2/t^2 x T)
C_{pi}	Species specific heat at constant pressure (L^2/t^2 x T x mole of i)
D	Diffusion coefficient (L^2/t)
D_i^T	Thermal diffusion coefficient (m/L x t)
$\mathbb{D}_{ij}$	Binary diffusion coefficient (L^2/t)
$\mathbb{E}$	Radiative flux divergence (m/L x t^2)
f	Velocity function defined by Eg. 4.14
f_j	Forward reaction rate constant for the jth reaction
F	Gibbs free energy (mL^2/t^2)
F_i	Species Gibbs free energy (mL^2/t^2 x mole of i)
H	Total enthalpy, $H = h + \frac{u^2}{2}$ (L^2/t^2)
h	Static enthalpy, $h = Q + P/\rho$ (L^2/t^2), also Planck's constant
h_i	Species static enthalpy (L^2/t^2 x mole of i)
I_v	Spectral radiation intensity (m/t^2 x no. of particles)

J_{iy}	Mass diffusion component for the i th species in a direction normal to the body surface ($m/L^2 \times t$)
k	Coefficient of thermal conductivity ($mL/t^3 \times T$)
K_j	Equilibrium constant for the jth reaction
M_i	Molecular weight of species i (mass of i/mole of i)
m	Number of chemical reactions occuring in the flow-field
m_i	Mass of i (m)
N	Number density (particles/L^3)
n	Number of chemical species present in the flow-field
n_i	Species molal density (moles of i/L^3)
P	Static pressure ($m/L \times t^2$) or (F/L^2)
Q	Internal energy per unit mass, including chemical energy (L^2/t^2), also total energy flux to a surface (m/t^3)
$q_{R,y}$	Radiative heat flux component in a direction normal to the body surface (m/t^3) or ($E/L^2 \times t$)
Qc	Convective energy flux to a surface (m/t^3)
Q_D	Diffusive energy flux to a surface (m/t^3)
Q_R	Radiative energy flux to a surface (m/t^3)
R	Body nose radius (L)
R_e	Reynolds number $\rho_{\infty,o} U_\infty R/\mu_{s,o}$

$R_{e\delta}$	Reynolds number $\rho_{s,o}$ U_∞ R/μ_{s_o}
IR	Universal gas constant (mL^3/t^2 x T x no. of moles)
r	Cylindrical body radius defined in Fig. 2.1 (L)
rj	Backward reaction rate constant for the jth reaction
S	Entropy (L^2/t^2 x T)
Si	Species entropy (L^2/t^2 x T x mole of i)
T	Thermodynamic temperature (T)
t	Time (t)
U_∞	Freestream velocity (L/t)
u	Velocity component in a direction parallel to the body surface (L/t)
v	Velocity component in a direction normal to the body surface (L/t) also frequency (l/t)
x	Body oriented coordinate defined in Fig. 2.1
y	Body oriented coordinate defined in Fig. 2.1
yi	Mole fraction of species i, $\sum_i yi = 1$

<u>Greek</u>

α_σ	Volumetric absorption coefficient, effective (L^2 x no. of particles/L^3)
δ	Shock detachment distance (L)

$\tilde{\delta}$	Transformed shock detachment distance
ε	Difference between the body and shock angle $\varepsilon = \theta - \phi$ radians
η	Dorodnitzyn variable
θ	Body angle (radians)
κ	Local body curvature (1/L)
$\tilde{\kappa}$	1 + y
Λ	Diffusional or radiative flux divergence (see Eq. 2.12) $(m/L\ z\ t^2)$
λ	$(\tilde{\mu} - 2/3\ \mu)$ (m/L x t)
μ	Ordinary viscosity (m/L x t)
$\tilde{\mu}$	Bulk viscosity (m/L x t)
ν	Frequency (1/t)
ξ	Nondimensional x-coordinate
ρ	Density (m/L^3)
ρi	Partial density of species i, $\rho i = n_i M_i$ (m of i/L^3)
$\bar{\rho}$	Density ratio across shock
ϕ	Shock angle (radians)
ωi	Generation of species i $(m/L^3 \times t)$

Subscripts

D	Diffusion
e	Edge conditions
i	Species i
n	Normal component
r	Radiation
t	Tangential component or total quantity
v	Spectral
w	Wall quantities
x	Component in a direction parallel to the ody surface
y	Component in a direction normal to the body surface
∞	Fresstream conditions
s, δ	Quantities immediately behind the shock

Superscripts

A	0 or 1 denoting two-dimensional or axisymetric respectively (an exponent)
T	Thermal
*	Denotes dimensional variables
o	Standard state quantity
-	Evaluated on char side of ablator interface
+	Evaluated on flow-field side of ablator interface

** Symbols not listed are defined where used.
$ Abbreviations mean: m, mass; L, length; t, time; T, temperature; F, force; E, energy

ABSTRACT

A mathematical model of the aerothermochemical environment along the stagnation line of a planetary return spacecraft using an ablative thermal protection system was developed and solved for conditions typical of atmospheric entry from planetary missions. The model, implemented as a Fortran IV computer program, was designed to predict viscous, reactive and radiative coupled shock layer structure and the resulting body heating rates. The analysis includes flow-field coupling with the ablator surface, binary diffusion, coupled line and continuum radiative and equilibrium or finite-rate chemistry effects. The gas model used includes thermodynamic, transport, kinetic and radiative properties of air and ablation product species, including 19 chemical species and 16 chemical reactions. Specifically, the impact of non-equilibrium chemistry effects upon stagnation line shock layer structure and body heating rates was investigated.

The mathematical model forms a set of twenty-six algebraic, differential and integrodifferential non-linear equations subject to two-point boundary conditions. The numerical solution was carried out by decoupling, linearizing and finite differencing the equations in a iterative, globally-implicit manner.

The model was used to determine the finite-rate chemistry and chemical-equilibrium stagnation line heating rate of a 9 foot entry vehicle protected by a phenolic-nylon ablator and moving at 50,000 feet/second when the free-stream air density is 8.85×10^{-8} slugs/ft^3 and the mass injection rate equals .05. The results obtained predict significantly different shock layer properties for the finite-rate

case and for the chemical equilibrium case. The ablation products entering the shock layer through the body wall react much slower than under equilibrium conditions and the air components entering the shock layer through the shock are not de-ionized, as predicted by the equilibrium analysis, but remain "frozen" throughout most of the shock region. The total heating rate to the body was found to be significantly lower under non-equilibrium than under equilibrium chemistry conditions. This difference was attributed to lower concentrations of optically active species, such as N and O, over most of the shock layer in the non-equilibrium chemistry case.

On the basis of this research it was concluded that a finite-rate stagnation-line shock layer solution which contains a reasonable kinetics model to describe atmospheric entries protected by phenolic-nylon ablators was successfully developed. It was also concluded that, for the flight conditions considered, finite-rate chemistry effects are significant since both shock layer structure and body heating rates are markedly different from those predicted by chemical equilibrium analyses. It was recommended that additional studies be carried out on: 1.) improving the computational speed of the solution; 2.) finite-rate chemistry effects under flight conditions different from those considered in this work; 3.) the effect of assuming more realistic shock and wall boundary conditions; and 4.) determining solutions for coupled non-equilibrium ablator response and shock layers. The computational speed of the solution should be improved so that SLAC is developed from a research program to an effective engineering tool. Studies of finite-rate chemistry effects under flight conditions different from those considered here should be carried out to determine the range of

conditions over which solutions may be obtained using the implemented model, and the range of conditions over which non-equilibrium chemistry effects are significant and how they affect the body heating rates. The boundary conditions studies should investigate the possible non-equilibrium composition of air, including detailed studies of the pertinent kinetics at the shock, (in the present work the chemical equilibrium composition was used), and the use of boundary conditions of the third kind at the wall. Coupled solutions for both non-equilibrium ablator behavior and shock layers should be determined to obtain a complete solution to the quasi-steady entry problem.

CHAPTER 1

INTRODUCTION

THE NATURE OF THE REENTRY PROBLEM

The launching of a suborbital, orbital or superorbital flight requires that some means of propulsion be utilized to propel the spacecraft against the force of gravity. In practice, a chemically-fueled rocket engine is used. The increase in the speed of the spacecraft results in an increase in its kinetic energy while its motion through the gravitational force field results in an increase in its potential energy. This kinetic and potential energy would be converted into the same amount of kinetic energy if the spacecraft were allowed to fall freely towards the Earth and there was no atmosphere. This would result in fantastic landing speeds and the destruction of the spacecraft upon its collision with the surface of the Earth. Obviously, additional propulsive power would have to be used in a direction opposite to the force of gravity in order to deccelerate the spacecraft until reasonable landing speeds are obtained. This need for additional propulsive power results in a decrease of the payload since some of the weight of the spacecraft has to be used for the propulsion hardware. N.A.S.A.'s Moon landings are accomplished precisely in this manner.

Decceleration of entry vehicles can be accomplished by: 1) use of propulsive power against the pull of gravity, and 2) use of aerodynamic drag. Landing of spacecrafts on celestial bodies lacking an atmosphere

precludes the use of the second technique. Aerodynamic braking requires transfer of the body's kinetic and potential energy to the atmosphere and results in the heating of the air in the vicinity of the body. Part of the energy gained by the air is then transferred to the body by convection and radiation creating a need for thermal protection of the spacecraft. Both the propulsive and aerodynamic drag braking techniques require a reduction of the payload since in the former the propulsion hardware and in the latter the heat shield must be a part of the spacecraft. The optimum technique to be used varies with mission characteristics and should be chosen according to a number of factors, such as the level of heat shielding technology and payload. Less payload penalty is suffered by using aerodynamic braking than propulsive braking for braking from hyperbolic approach velocities to orbital velocities at Mars (Ref. 1.1).

DESCRIPTION OF THE FLOW-FIELD IN THE VICINITY OF THE BODY

Of primary importance, in establishing entry heating levels, is the magnitude of the speeds that will be encountered. The range of Earth entry speeds associated with several mission objectives is given in Figure 1.1 (Ref. 1.2). The use of the gravitational field of Venus to alter the interplanetary trajectories between Mars and Earth (Venus swingby) is shown to result in significantly reduced Earth entry speeds when compared to direct trajectories. Of particular interest are entry speeds of about 50,000 fps (about 15 km/sec) since the capability of entering at these speeds will permit a wide variety of trips to Mars, Venus and the asteroids.

A blunt body of the type used by NASA in its manned spacecraft program (Figure 1.2) would show, under conditions prevalent during atmospheric entry at speeds of approximately 50,000 fps, a standing bow

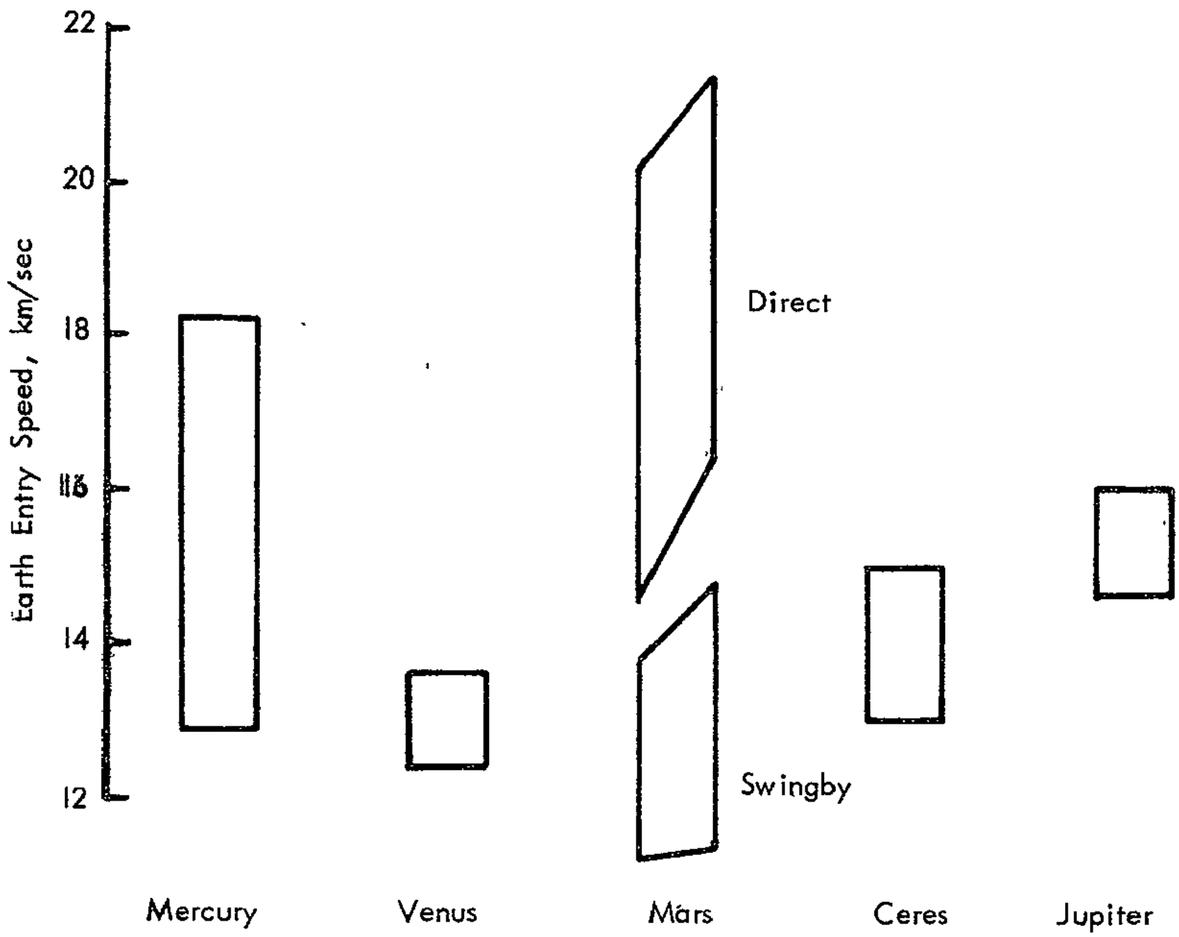


Figure 1.1 Earth Entry Speeds for Several Mission Objectives
(From Ref. 1.2)

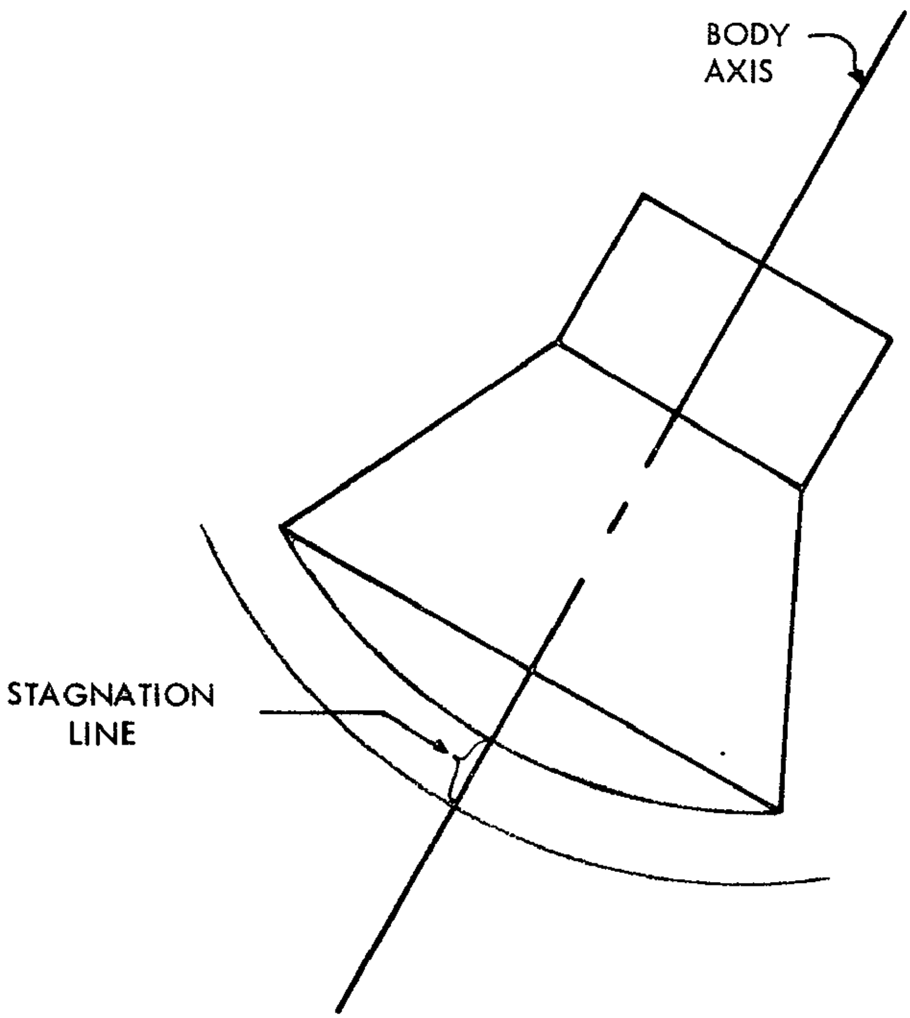


Figure 1.2 Flow-Field in the Vicinity of a Blunt Body

shock due to the decceleration of the air in the vicinity of the body. The blunt body of Figure 1.2 is axisymmetric and is shown flying at a zero angle of attack (its axis of symmetry coincides with the translational vector), this means that the shock and the flow field between the shock and the body are axisymmetric. The region of the flow field between the shock and the surface of the body may be divided into the stagnation line, so-called because the velocity is zero somewhere along this line, and the around-the-body region. When the body wall is impermeable, the stagnation point is at the wall where both the normal and tangential components of velocity are zero.

It is important to note, that even though viscous effects are in part responsible for the generation of the high temperatures present, they are not solely responsible since even for inviscid fluids there would be a decceleration of the fluid at the stagnation line. Thus the difference between viscous and inviscid flow fields would not be basic in nature.

THERMAL PROTECTION SYSTEMS

The task of protecting the vehicle from the heat generated during atmospheric entry may be accomplished in a number of ways: use of heat sinks, transpiration cooling, radiation cooling, ablative cooling, etc. The use of a heat sink as a thermal protection system is very limited by the lack of suitable materials with high heat capacity and low thermal conductivity. Transpiration cooling involves the injection of a cool fluid into the shock layer so that it absorbes heat and insulates the vehicle from the high temperatures. Radiation cooling involves the use of a highly reflective structure in order to block the transfer of radiative heat into the body. Ablative heat shields operate with materials that undergo endothermic physical and chemical changes.

Of all the methods of thermal protection, ablative cooling has been the most successful. This type of thermal protection system may be either a non-charring or a charring ablator. The former type of ablator is one that changes from a solid to a gaseous state and enters the shock layer counter to the heat flow; typical of these materials is Teflon. Charring ablators, on the other hand, decompose to a porous char and relatively low molecular weight gases. The latter type ablator has enjoyed widespread use in the space program.

The principal thermal protection mechanisms in a charring ablator are shown in Figure 1.3. Blockage occurs when the pyrolisis gases are injected into the shock layer and prevent most of the convective heat flux from reaching the body surface. The ablation products in the shock layer absorb part of the radiative flux from the shock and also react endothermically. Char re-radiation stops part of the radiative flux to the body while, if the temperature is high enough, sublimation of the char absorbs part of the incoming energy. The further "cracking" of pyrolisis gases and the process of heating those gases from the temperature at the decomposition zone to the shock layer temperature and their subsequent injection into the shock layer constitute additional protection mechanisms. The decomposition process in the pyrolisis zone which initially produces the pyrolisis gas also absorbs heat. Finally, any heat not absorbed elsewhere may be stored in the virgin material in which case the process acts as a simple heat sink.

THE NEED FOR A BETTER UNDERSTANDING OF THE AERODYNAMIC BRAKING PROCESS

Results from a study of the possible effects of uncertainties in analyses upon heat shield weight are given in Fig. 1.4 (Ref. 1.2). The factors considered were uncertainties in: air absorption coefficients,

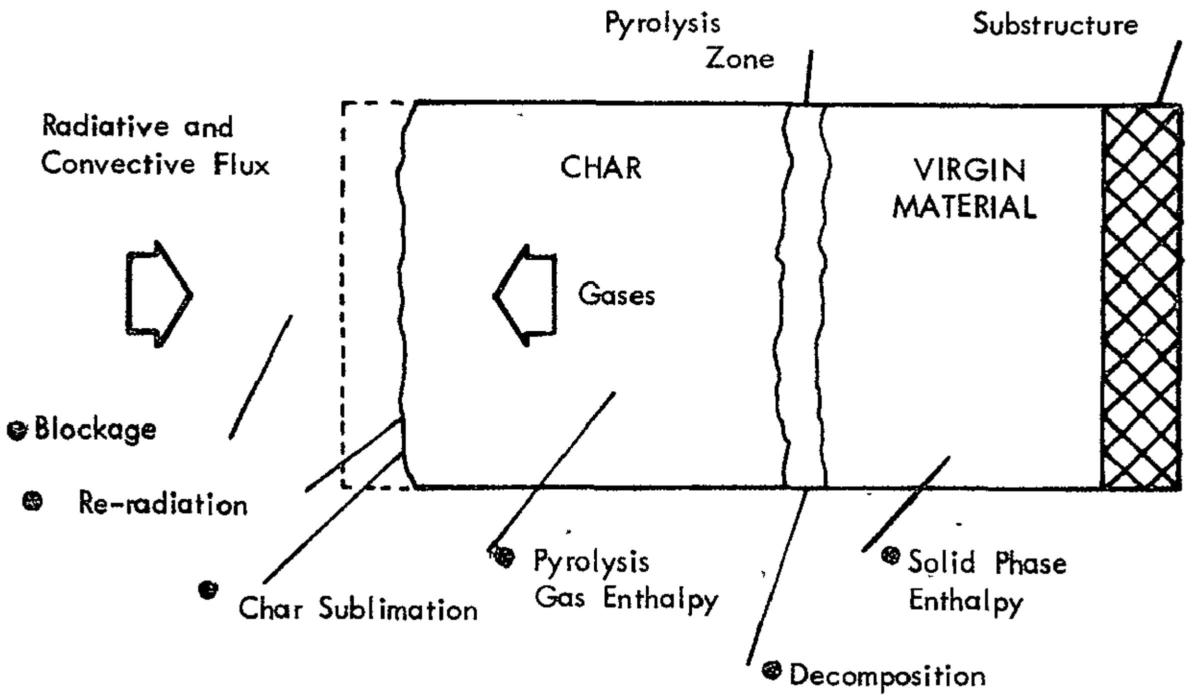


Figure 1.3 Thermal Protection Mechanisms (From Ref. 1.2)

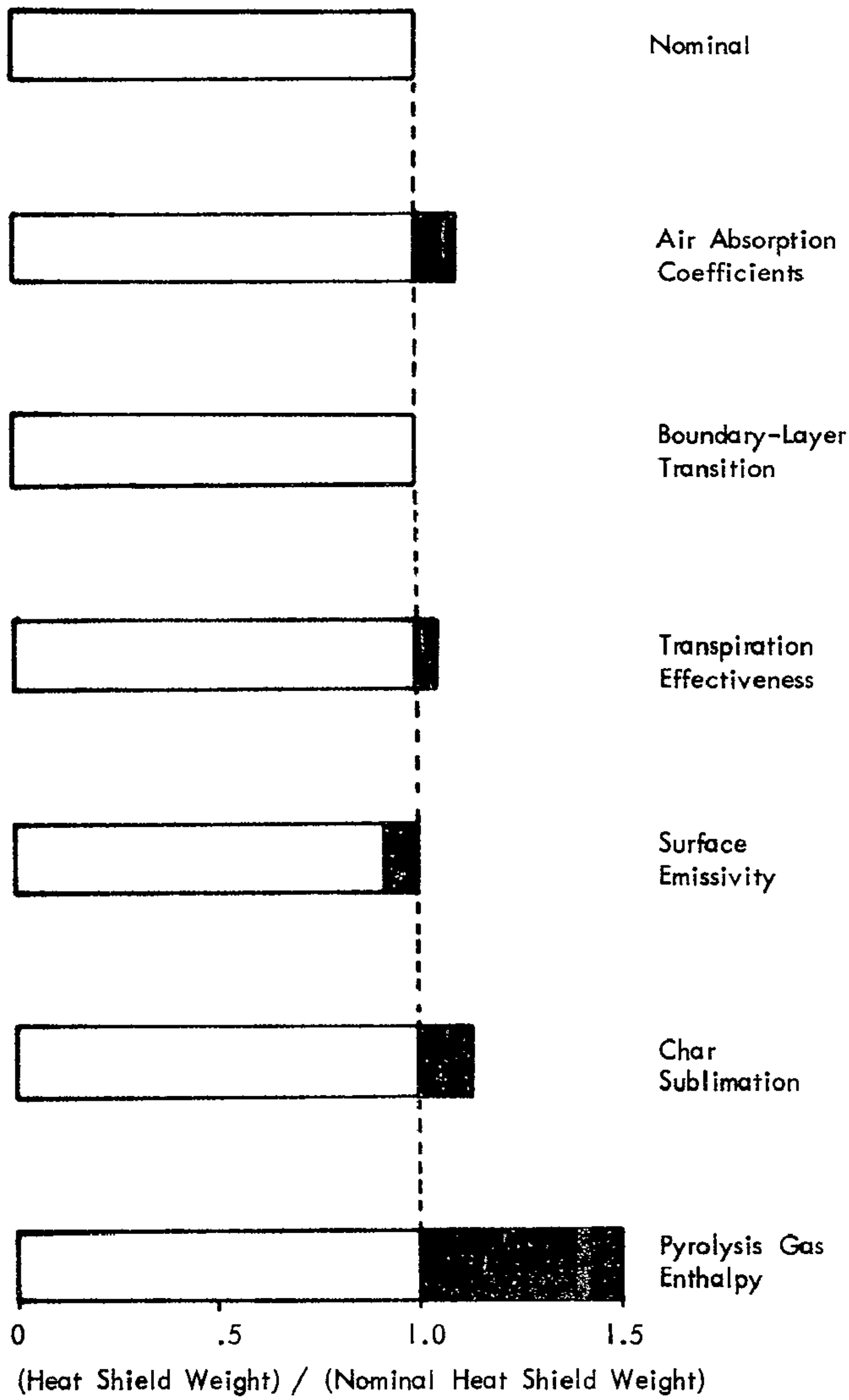


Figure 1.4 Results of Uncertainty Analysis for an Entry Speed of 15 km/sec (From Ref. 1.2)

boundary layer transition, transpiration effectiveness, surface emissivity, char sublimation, and pyrolisis gas enthalpy. The entry speed considered is 50,000 fps and the weights are normalized to those for the nominal conditions as shown at the top. Results for each perturbation are stated. It should be noted that the only perturbation causing a decrease in heat shield weight is the surface emissivity. Since the effect of some of the uncertainties can be additive it is evident that a weight increase of 50 percent or more is possible. For the range of entry speeds of interest in this study the heat shield is estimated to represent between 10 and 20 percent of the entry-vehicle weight. Due to the uncertainties mentioned above this value can represent as high as 15 to 30 percent of the entry vehicle weight with a corresponding reduction in the payload returnable to Earth. Such an effect is clearly not negligible.

One obvious way of reducing these uncertainties is by experimental verification of the thermal environment about the spacecraft. Such a verification requires duplication of a number of parameters like: flight velocity, free-stream density, enthalpy and flow energy, etc. Unfortunately, the presently existing ground testing facilities are not capable of simultaneous duplication of all the important parameters (Refs. 1.2, 1.4). The alternatives to ground testing are: testing under actual flight conditions or developing accurate mathematical models to describe the aerodynamic braking process. Because of the very high costs involved in flight testing, the importance of improving existing analytic models is evident.

The pyrolisis gas enthalpy uncertainty, being the largest has since being studied (Ref. 1.4). For entry speeds greater than 50,000

fps, the effect of the radiation heat transfer uncertainty should become increasingly important since at these higher speeds the ratio of heat transfer by radiation to heat transfer by convection increases. A recent study by Engel (Ref. 1.5) resulted in development of a model that includes line and continuum radiation and equilibrium chemistry with no diffusion. Each (Ref. 1.6) used the same radiation model used by Engel and obtained solutions for binary and multicomponent diffusion of gas mixtures in chemical equilibrium

Radiative properties are dependent, among other things, upon chemical composition. For this reason it is important to use the correct chemical model when computing heating rates. The chemical equilibrium assumption greatly simplifies the problem since it eliminates the need for consideration of chemical kinetics. When conditions in the flowfield are such that the actual chemical composition is close to the chemical equilibrium composition, then the use of an equilibrium model is valid. This generally occurs when the pressure and temperature are high. On the other hand for low pressures and temperatures the difference between the actual and equilibrium compositions may be large and a finite-rate chemistry model must be used.

STATEMENT OF OBJECTIVES

This study was undertaken in an effort to reduce uncertainties in reentry heating resulting from non-equilibrium chemistry effects. The state-of-the art upon which improvements will be made is given in Refs. 1.5 and 1.6. Specifically, the current research will try to fulfill the following objectives:

1. Develop and solve a mathematical model of the flow-field along the stagnation-line of the shock layer. The model shall

include the effects of line and continuum radiation, thermodynamic and transport properties of air and ablation products, binary diffusion, and finite-rate chemistry.

2. Determine whether finite-rate chemistry effects significantly change the heating rate from the one obtained by assuming the flow is in chemical equilibrium.

REFERENCES

1.1 Eggers, Alfred J., and Nathaniel B. Cohen, "Progress and Problems in Atmosphere Entry," Proceedings of the XVIth International Astronautical Congress, Gauthier-Villars, Paris, 1965.

1.2 Syvertson, C.A., "Research Problems in Atmosphere Entry and Landing for Manned Planetary Missions," NASA TN D-4977, January 1967.

1.3 Steward, J.D., and D.H. Greenshields, "Entry Vehicles for Space Programs," Journal of Spacecraft and Rockets, Vol. 6, No. 10, October 1969, pp. 1089-1102.

1.4 Gonzalez del Valle, Eduardo, In Depth Response of Ablative Composites, Ph.D. Dissertation, Louisiana State University, Baton Rouge, Louisiana (1974).

1.5 Engel, Carl D., Ablation and Radiation Coupled Viscous Hypersonic Shock Layers, Ph.D. Dissertation, Louisiana State University, Baton Rouge, Louisiana (1971)

1.6 Esch, Donald D., Stagnation Region Heating of a Phenolic-Nylon Ablator During Return from Planetary Missions, Ph.D. Dissertation, Louisiana State University, Baton Rouge, Louisiana (1971).

CHAPTER 2

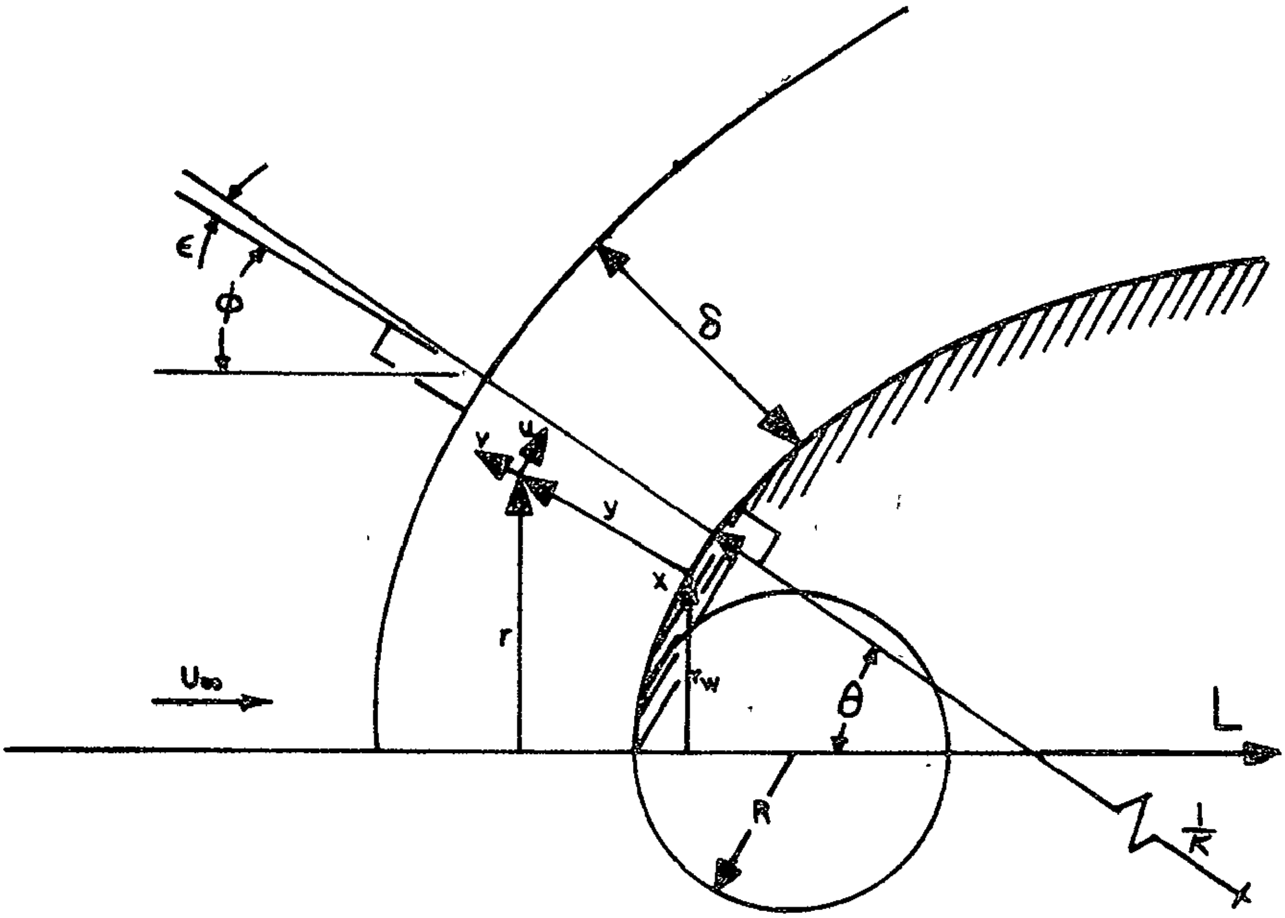
DEVELOPMENT OF THE FLOW-FIELD EQUATIONS

The mathematical model which describes the flow-field at the stagnation line is developed in this chapter. The conservation equations for the flow-field between the body and the shock with the appropriate wall and shock boundary conditions are developed. From these equations, the stagnation line conservation equations and boundary conditions are derived.

CONSERVATION EQUATIONS OF THE SHOCK LAYER

The mathematical model of the flowfield from the body surface to the shock is obtained from the laws of conservation of mass, momentum and energy and also from the formulation of a gas model. The gas model used will be discussed in the latter part of this chapter as well as in Chapter 3. The complete body oriented conservation equations for a multicomponent, viscous, radiating, chemically reacting fluid at steady state were developed by Engel (Ref. 2.1). The body oriented coordinate system and shock layer geometrical relations pertinent to these conservation equations are given in Figure 2.1.

However, these equations are far too complex to allow solution with presently available techniques. Fortunately, it has been found that, in general, solution of the complete conservation equations is not necessary since some of the terms in the equations do not significantly contribute to the solution and may therefore be neglected. Of



$$\operatorname{Tan} \epsilon = \frac{d\delta}{(1+\kappa\delta)\,dx} \qquad \frac{d\theta}{dx} = \kappa(x)$$

$$\delta = \int_0^x (1+\kappa\delta) \operatorname{Tan} \epsilon \, dx + \delta_0 \qquad \theta = \int_0^x \kappa(x)\,dx$$

Figure 2.1 Body-Oriented Coordinate System

course, which terms, if any, are negligible varies with flow conditions and must be determined in each particular case. It will be shown below that, for the flight conditions of interest, the shock layer is laminar, thin is a continuum and these facts can be used to obtain the order of magnitude of the terms in the conservation equation.

According to the results of Hayes and Probstein (Ref. 2.2) the gas behind a bow shock of a hypervelocity vehicle is a continuum when the free-stream Reynolds number (Re) satisfies the relation

$$R_e = \frac{\rho_\infty{}^* U_\infty{}^* R^*}{\mu^*_{\delta,o}} > 100 \qquad (2.1)$$

Further, for free-stream Reynolds numbers greater than 100, the standoff distance divided by body radius (δ^*/R^*) has been shown to be approximately equal to the ratio of the pre-shock density (ρ^*_∞) to the post-shock density (ρ^*_δ) (Ref. 2.2).

$$\frac{\delta^*}{R^*} \simeq \frac{\rho_\infty}{\rho_\delta{}^*} = \bar{\rho} \qquad (2.2)$$

Finally, for hypersonic Mach numbers the density ratio across the shock ($\bar{\rho}$) is of the order of one tenth and less for dissociating gases.

$$\bar{\rho} \leq .10 \qquad (2.3)$$

An estimate of the dimensionless standoff distance can be obtained from Eqs. 2.1 and 2.2.

$$\frac{\delta^*}{R^*} \leq .10 \qquad (2.4)$$

In this chapter a superscript * will denote dimensional variables unless it is explicitly stated otherwise.

Since the dimensionless distance in a direction parallel to the body (X^*/R^*) is of order 1 and greater while the dimensionless distance in a direction normal to the body is of order $\frac{1}{10}$ and less, this means that the shock layer is thin.

In order to determine the order of magnitude of the terms in the conservation equations following the methods of Ref. 2.3, the equations are nondimensionalized by using constants characteristic of the flow field, for example

$$\xi = \frac{X^*}{R^*}$$

$$y = \frac{y^*}{R^*} \qquad (2.5)$$

$$u = \frac{u^*}{U_\infty^*}$$

$$v = \frac{v^*}{U_\infty^*}$$

The order of magnitude of the dependent and independent variables is determined by using the largest possible magnitude of the dimensional variable, for example,

$$\xi \sim \frac{R^*}{R^*} = 1$$

$$y \sim \frac{\delta^*}{R^*} = \bar{\rho} \qquad (2.6)$$

$$u \sim \frac{U_\infty^*}{U_\infty^*} = 1$$

Using the above procedure, the orders of magnitude of the terms in the conservation equations were determined (Ref. 2.1) and are given in Table 2.1. All terms of order $\bar{\rho}^2$ and higher have been dropped from all equations except the y-momentum equation. The dimensionless variables appearing in Eqs. 2.7 - 2.12 are defined in Table 2.2. It must be noted that Eqs. 2.8, 2.9 and 2.10 contain terms of unknown order of magnitude.

TABLE 2.1

LISTING OF CONSERVATION EQUATION WITH

ORDER ASSESSMENT RESULTS

Global continuity:

$$\underbrace{\frac{\partial}{\partial \xi}(r^A \rho u)}_{O[1]} + \underbrace{\frac{\partial}{\partial y}(\tilde{\kappa} r^A \rho v)}_{O[1]} = 0 \qquad (2.7)$$

Species continuity:

$$\underbrace{\frac{\partial}{\partial \xi}(r^A \rho C_i u)}_{O[1]} + \underbrace{\frac{\partial}{\partial y}(\tilde{\kappa} r^A \rho C_i v)}_{O[1]} = \underbrace{- \frac{\partial}{\partial y}(\tilde{\kappa} r^A J_{iy})}_{O[1]} + \tilde{\kappa} r^A \omega_i \qquad (2.8)$$

ξ - momentum:

$$\underbrace{\rho r^A u \frac{\partial u}{\partial \xi}}_{O[1]} + \underbrace{\rho \tilde{\kappa} r^A v \frac{\partial u}{\partial y}}_{O[1]} + \underbrace{\rho \kappa r^A uv}_{O[\bar{\rho}]} + r^A \frac{\partial P}{\partial \xi}$$

$$- \frac{1}{R_e} \left\{ \underbrace{\frac{\partial}{\partial y} \left(\tilde{\kappa} r^A \mu \frac{\partial u}{\partial y} \right)}_{O\left[\frac{1}{\bar{\rho}^2}\right]} - \underbrace{\frac{\partial}{\partial y} (\kappa r^A \mu u)}_{O\left[\frac{1}{\bar{\rho}}\right]} + \underbrace{\kappa r^A \mu \frac{\partial u}{\partial y}}_{O\left[\frac{1}{\bar{\rho}}\right]} \right\} = 0 \qquad (2.9)$$

TABLE 2.1 (Cont.)

y - momentum: ($O[\bar{\rho}]$ and larger terms)

$$
\begin{aligned}
&\overset{O[\bar{\rho}]}{\rho r^A u \frac{\partial v}{\partial \xi}} + \overset{O[\bar{\rho}]}{\rho r^A \tilde{\kappa} v \frac{\partial v}{\partial y}} - \overset{O[1]}{\rho \tilde{\kappa} r^A u^2} + \tilde{\kappa} r^A \frac{\partial P}{\partial y} \\
&- \frac{1}{R_e} \left\{ \overset{O\left[\frac{1}{\bar{\rho}}\right]}{\frac{\partial}{\partial \xi}\left(r^A \mu \frac{\partial u}{\partial y}\right)} + \overset{O\left[\frac{1}{\bar{\rho}}\right]}{\frac{\partial}{\partial y}\left(\lambda \frac{\partial}{\partial \xi}(r^A u)\right)} \right. \\
&\left. + \overset{O\left[\frac{1}{\bar{\rho}}\right]}{\frac{\partial}{\partial y}\left(\lambda \frac{\partial}{\partial y}(\tilde{\kappa} r^A v)\right)} + \overset{O\left[\frac{1}{\bar{\rho}}\right]}{\frac{\partial}{\partial y}\left(2 \tilde{\kappa} r^A \mu \frac{\partial v}{\partial y}\right)} \right\} = 0
\end{aligned}
\tag{2.10}
$$

y - momentum: ($O[\bar{\rho}^2]$ and larger terms)

$$
\begin{aligned}
&\overset{O[\bar{\rho}]}{\cancelto{0}{\rho r^A u \frac{\partial v}{\partial \xi}}} + \overset{O[\bar{\rho}]}{\rho r^A \tilde{\kappa} v \frac{\partial v}{\partial y}} - \overset{O[1]}{\cancelto{0}{\rho \kappa r^A u^2}} + \tilde{\kappa} r^A \frac{\partial P}{\partial y} \\
&- \frac{1}{R_e} \left\{ \overset{O\left[\frac{1}{\bar{\rho}}\right]}{\frac{\partial}{\partial \xi}\left(r^A \mu \frac{\partial u}{\partial y}\right)} + \overset{O\left[\frac{1}{\bar{\rho}}\right]}{\frac{\partial}{\partial y}\left(\lambda \frac{\partial}{\partial \xi}(r^A u)\right)} \right. \\
&+ \overset{O\left[\frac{1}{\bar{\rho}}\right]}{\frac{\partial}{\partial y}\left(\lambda \frac{\partial}{\partial y}(\tilde{\kappa} r^A v)\right)} + \overset{O\left[\frac{1}{\bar{\rho}}\right]}{\frac{\partial}{\partial y}\left(2 \tilde{\kappa} r^A \mu \frac{\partial v}{\partial y}\right)} - \overset{O[1]}{\frac{\partial}{\partial \xi}(r^A \mu \kappa u)} \\
&\left. - \overset{O[1]}{\frac{\lambda \kappa}{\tilde{\kappa}} \frac{\partial}{\partial \xi}(r^A u)} - \overset{O[1]}{\frac{\lambda \kappa}{\tilde{\kappa}} \frac{\partial}{\partial y}(\tilde{\kappa} r^A v)} - \overset{O[1]}{2\mu \frac{\kappa r^A}{\tilde{\kappa}} \frac{\partial u}{\partial \xi}} \right\} = 0
\end{aligned}
\tag{2.11}
$$

TABLE 2.1 (Cont.)

Energy:

$$
\begin{aligned}
&\overset{O[1]}{r^A \rho u \frac{\partial H}{\partial \xi}} + \overset{O[1]}{\tilde{\varkappa}\, r^A \rho v \frac{\partial H}{\partial y}} = -\, 2\Big(\overset{O[1]}{\Lambda_{D,y}} + \overset{O[1]}{\Lambda_{R,y}} \Big) \\
&+ \frac{2}{R_e} \left\{ \overset{O\left[\frac{1}{\rho^2}\right]}{\frac{\partial}{\partial y} \left(\tilde{\varkappa} r^A \mu u \frac{\partial u}{\partial y} \right)} - \overset{O\left[\frac{1}{\rho}\right]}{\frac{\partial}{\partial y} \left(\varkappa r^A \mu u^2 \right)} \right\}
\end{aligned}
\qquad (2.12)
$$

TABLE 2.2

DEFINITION OF DIMENSIONLESS VARIABLES

$$\xi = \frac{x^*}{R^*} \qquad y = \frac{y^*}{R^*} \qquad u = \frac{u^*}{U_\infty^*} \qquad v = \frac{v^*}{U_\infty^*}$$

$$\rho = \frac{\rho^*}{\rho_{\infty,0}^*} \qquad \mu = \frac{\mu^*}{\mu_{s,0}^*} \qquad \lambda = \frac{\lambda^*}{\mu_{s,0}^*} \qquad \delta = \frac{\delta^*}{R^*}$$

$$r = \frac{r^*}{R^*} \qquad \varkappa = \varkappa^* R^* \qquad P = \frac{P^*}{\rho_\infty^*(U_\infty^*)^2} \quad H = \frac{H^*}{H_s^*}$$

$$h = \frac{h_s^*}{H_s^*} \qquad h = \frac{h^*}{h_s^*} \quad \text{where} \quad H_s^* = \tfrac{1}{2} U_\infty^{*2}$$

$$\tilde{\varkappa} = 1 + \varkappa y \qquad \omega_i = \frac{R^* \omega_i^*}{\rho_\infty^* U_\infty^*} \qquad J_i = \frac{J_i^*}{\rho_\infty^* U_\infty^*} \qquad \Lambda_{R,x} = \frac{\Lambda^*_{R,x}}{\rho_\infty^*(U_\infty^*)^3}$$

$$\Lambda_{R,y} = \frac{\Lambda^*_{R,y}}{\rho_\infty^*(U_\infty^*)^3} \qquad \Lambda_{D,x} = \frac{\Lambda^*_{D,x}}{\rho_\infty^*(U_\infty^*)^3} \qquad \Lambda_{D,y} = \frac{\Lambda^*_{D,y}}{\rho_\infty^*(U_\infty^*)^3}$$

where

$$\Lambda^*_{R,x} = \frac{\partial}{\partial x^*}(r^{*A} q^*_{R,x}) \qquad \Lambda^*_{R,y} = \frac{\partial}{\partial y^*}(r^{*A} \tilde{\varkappa} q^*_{R,y})$$

$$\Lambda^*_{D,x} = \frac{\partial}{\partial x^*}(r^{*A} q^*_{D,x}) \qquad \Lambda^*_{D,y} = \frac{\partial}{\partial y^*}(r^{*A} \tilde{\varkappa} q^*_{D,y})$$

This means that these terms can not be dropped from the equations since they may or may not be significant.

Two additional assumptions were made in order to simplify the equations of the flow-field. The first of these consists of assuming that Stokes' Postulate (Ref. 2.3) is valid throughout the flow-field. This yields

$$\lambda^* = -\frac{2}{3}\ \mu^* \tag{2.13}$$

and simplifies Eqs. 2.10 and 2.11. The radiative energy term in Eq. 2.12 was simplified by assuming that the shock layer geometry is approximated locally by an infinite plane slab in which radiative transport properties vary only across the slab. This assumption yields

$$\begin{aligned}\Lambda_{Ry} &= \frac{\partial(\tilde{\kappa}\ r^A\ q_{R,y})}{\partial\ y} \\ &= -\ 2\pi\ \tilde{\kappa}\ r^A \int_0^\infty \alpha_\nu\ (2B_\nu - I_\nu)\ d\nu \end{aligned}\tag{2.14}$$

As a result of the bulk viscosity assumption and the planar radiative transfer model Eqs. 2.7-2.12 may be written in a more usable form. The resulting equations are known as the thin shock layer equations and are given in Table 2.3. These equations are also referred to as the second order boundary layer equations with curvature effects.

If all the terms of order $\bar{\rho}$ or higher are dropped from the conservation equations they are then known as the first order shock layer equations or the first order boundary layer equations (Table 2.4).

The first order boundary layer equations (Table 2.4) were chosen to describe the flowfield for the following reasons: (1) the Reynolds numbers encountered during Earth atmospheric reentry from typical Mars

TABLE 2.3

SECOND ORDER SHOCK LAYER EQUATIONS

Global continuity:

$$\frac{\partial}{\partial x}(r^A \rho u) + \frac{\partial}{\partial y}(\tilde{\varkappa} r^A \rho v) = 0 \tag{2.15}$$

Species continuity:

$$\frac{\partial}{\partial x}(r^A \rho C_i u) + \frac{\partial}{\partial y}(\tilde{\varkappa} r^A \rho C_i v) = -\frac{\partial}{\partial y}(\tilde{\varkappa} r^A J_{i,y}) + \tilde{\varkappa} r^A \omega_i \tag{2.16}$$

x - Momentum:

$$\rho r^A u \frac{\partial u}{\partial x} + \rho \tilde{\varkappa} r^A v \frac{\partial u}{\partial y} + \rho \varkappa r^A uv = - r^A \frac{\partial P}{\partial x}$$

$$+ \frac{\partial}{\partial y}(\tilde{\varkappa} r^A \mu \frac{\partial u}{\partial y}) - \varkappa u \frac{\partial r^A \mu}{\partial y} \tag{2.17}$$

y - Momentum: $(O[\bar{\rho}]$ and larger terms)

$$\rho r^A u \frac{\partial v}{\partial x} + \rho \tilde{\varkappa} r^A v \frac{\partial v}{\partial y} - \rho \varkappa r^A u^2 = - \tilde{\varkappa} r^A \frac{\partial P}{\partial y}$$

$$+ \frac{\partial}{\partial x}(r^A \mu \frac{\partial u}{\partial y}) - \frac{2}{3}\frac{\partial}{\partial y}(\mu \frac{\partial r^A u}{\partial x}) + \frac{4}{3}\frac{\partial}{\partial y}(\varkappa r^A \mu \frac{\partial v}{\partial y}) \tag{2.18}$$

$$- \frac{2}{3}\frac{\partial}{\partial y}(\varkappa r^A \mu v + \tilde{\varkappa} \mu v \frac{\partial r^A}{\partial y})$$

TABLE 2.3 (Cont.)

y - Momentum: (0 $[\bar{\rho}^2]$ and larger terms)

$$\rho r^A u \frac{\partial v}{\partial x} + \rho r^A \tilde{\varkappa} v \frac{\partial v}{\partial y} - \rho \varkappa r^A u^2 = - \tilde{\varkappa} r^A \frac{\partial P}{\partial y}$$

$$+ \frac{\partial}{\partial x} (r^A \mu \frac{\partial u}{\partial y}) - \frac{2}{3} \frac{\partial}{\partial y} (\mu \frac{\partial r^A u}{\partial x}) + \frac{4}{3} \frac{\partial}{\partial y} (\tilde{\varkappa} r^A \mu \frac{\partial v}{\partial y})$$

$$- \frac{2}{3} \frac{\partial}{\partial y} (\varkappa r^A \mu v) - \frac{\partial}{\partial x} (r^A \varkappa \mu u) + \frac{2}{3} \mu \frac{\varkappa}{\tilde{\varkappa}} \frac{\partial}{\partial x} (r^A u) \tag{2.19}$$

$$+ \frac{2}{3} \mu \frac{\varkappa}{\tilde{\varkappa}} \frac{\partial}{\partial y} (\tilde{\varkappa} r^A v) - 2\mu \frac{\varkappa}{\tilde{\varkappa}} r^A \frac{\partial u}{\partial x} - \frac{2}{3} \frac{\partial}{\partial y} (\tilde{\varkappa} \mu v \frac{\partial r^A}{\partial y})$$

Energy:

$$r^A \rho u \frac{\partial H}{\partial x} + \tilde{\varkappa} r^A \rho v \frac{\partial H}{\partial y} = \frac{-\partial}{\partial y} \left[\tilde{\varkappa} r^A \left\{ -k \frac{\partial T}{\partial y} + \sum_i h_i J_{i,y} \right. \right.$$

$$\left. \left. - \frac{P}{N^2} \sum_i \sum_{j \neq i} \frac{N_i}{m_i} \frac{D_i^T}{\mathbb{D}_{ij}} \left(\frac{J_{j,y}}{\rho_j} - \frac{J_{i,y}}{\rho_i} \right) \right\} \right] - \tilde{\varkappa} r^A \frac{\partial q_{R,y}}{\partial y} \tag{2.20}$$

$$\frac{\partial}{\partial y} (\tilde{\varkappa} r^A \mu u \frac{\partial u}{\partial y}) - \frac{\partial}{\partial y} (\varkappa r^A \mu u^2)$$

TABLE 2.4

FIRST ORDER SHOCK LAYER EQUATIONS

(Order determined at $R_e = \rho_\infty U_\infty R/\mu_{\delta,o} = 100$)

Global continuity:

$$\frac{\partial}{\partial x}(\rho r^A u) + \frac{\partial}{\partial y}(\rho \tilde{\varkappa} r^A v) = 0 \qquad (2.21)$$

Species continuity:

$$\frac{\partial}{\partial x}(r^A \rho C_i u) + \frac{\partial}{\partial y}(\tilde{\varkappa} r^A \rho C_i v) = -\frac{\partial}{\partial y}(\tilde{\varkappa} r^A J_{i,y}) + \tilde{\varkappa} r^A \omega_i \qquad (2.22)$$

x - Momentum:

$$\rho r^A u \frac{\partial u}{\partial x} + \rho \tilde{\varkappa} r^A v \frac{\partial u}{\partial y} = -r^A \frac{\partial P}{\partial x} + \frac{\partial}{\partial y}\left[\tilde{\varkappa} r^A \mu \frac{\partial u}{\partial y}\right] \qquad (2.23)$$

y - Momentum:

$$\rho \varkappa u^2 = \tilde{\varkappa} \frac{\partial P}{\partial y} \qquad (2.24)$$

Energy:

$$r^A \rho u \frac{\partial H}{\partial x} + \tilde{\varkappa} r^A \rho v \frac{\partial H}{\partial y} = \frac{\partial}{\partial y}\left(\tilde{\varkappa} r^A k \frac{\partial T}{\partial y}\right) - \frac{\partial}{\partial y}\left(\tilde{\varkappa} r^A \left\{\sum_i h_i J_{i,y}\right.\right.$$

$$\left.\left. - \frac{P}{N^2} \sum_i \sum_{j \neq i} \frac{N_i}{m_i} \frac{D_i^T}{\mathbb{D}_{i,j}} \left(\frac{J_{j,y}}{\rho_j} - \frac{J_{i,y}}{\rho_i}\right)\right\}\right) - \tilde{\varkappa} r^A \frac{\partial q_{R,y}}{\partial y} \qquad (2.25)$$

$$+ \frac{\partial}{\partial y}\left(\tilde{\varkappa} r^A \mu u \frac{\partial u}{\partial y}\right)$$

are of the order of 10^4* making second order terms even less significant than was shown by the order of magnitude analysis; (2) there are inherent approximations in the theoretical models and experimental data used to predict thermodynamic, transport, radiation and chemical kinetics properties; therefore, there is no need for more detail in the equations solved than in the properties used in the equations.

Further simplification of the conservation equations was obtained by noticing that for the flight conditions and body radii of interest in this work the shock layer is very thin, therefore the relation $\tilde{\kappa} = 1 + \kappa y$ may be approximated by $\tilde{\kappa} = 1$. In addition, neglecting thermal diffusion yields the so-called boundary layer equations (Table 2.5).

BOUNDARY CONDITIONS

As was mentioned above, the boundary layer equations are parabolic and thus require the specification of boundary conditions on the stagnation line ($x = 0$), wall ($y = 0$) and shock ($y = \delta$). The stagnation line boundary conditions are obtained by taking the limit of Eqs. 2.26 - 2.30 as $x \to 0$. The resulting equations form the basis for most of the work performed in this dissertation and they will be developed in the next section. Wall and shock boundary conditions will be discussed below.

Wall Boundary Conditions

The development of wall boundary conditions may be performed in two different ways. The first, and most frequently used technique

---

*For typical entry conditions of $U_\infty^* = 50{,}000$ fps, $\rho_\infty^* = 10^{-5}$ lbm/ft^3 (corresponding to an altitude of 214,000 ft (Ref. 2.4)) and $R^* = 10$ft, a value of $T_\delta^* = 15{,}000$ °K is obtained, therefore $\mu_{\delta,o} = 10^{-4}$ lbm/ft sec (Ref. 2.5) and $Re = \dfrac{\rho_\infty^* R^* U_\infty^*}{\mu_{\delta,o}^*} \simeq 5 \times 10^4$.

TABLE 2.5

BOUNDARY LAYER EQUATIONS

Global continuity:

$$\frac{\partial}{\partial x} (r_w{}^A \rho u) + r_w{}^A \frac{\partial}{\partial y} (\rho v) = 0 \qquad (2.26)$$

Species continuity:

$$\frac{1}{r_w{}^A} \frac{\partial}{\partial x} (r_w{}^A \rho C_i u) + \frac{\partial}{\partial y} (\rho C_i v) = - \frac{\partial}{\partial y} (J_{i,y}) + \omega_i \qquad (2.27)$$

x - Momentum:

$$\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = - \frac{\partial P}{\partial x} + \frac{\partial}{\partial y} (\mu \frac{\partial u}{\partial y}) \qquad (2.28)$$

y - Momentum:

$$0 = \frac{\partial P}{\partial y} \qquad (2.29)$$

Energy:

$$\rho u \frac{\partial H}{\partial x} + \rho v \frac{\partial H}{\partial y} = \frac{\partial}{\partial y} (k \frac{\partial T}{\partial y}) - \frac{\partial}{\partial y} \left\{ \sum_i h_i J_{i,y} \right. - \frac{\partial q_{R,y}}{\partial y} \qquad (2.30)$$

$$+ \frac{\partial}{\partial y} (\mu u \frac{\partial u}{\partial y})$$

consists of writing mass, momentum, and energy balances across the ablator-shock layer interface (Fig. 2.2). A second technique involves the use of the conservation equations of the flow-field which are integrated from y^+ to y^- and then the resulting equations are evaluated as $\Delta y \to 0$. Both of these methods should yield identical results since the conservation equations used in the second technique are mass, momentum, and energy balances at any point in the flow-field. The first method has the advantage that the physical significance of each term in the resulting equations is more readily evident. The second technqiue, on the other hand will assure that all the necessary terms have been considered.

In this work the second technique discussed above was applied to the boundary layer equations (Eqs. 2.26 - 2.30) in order to obtain the surface balances needed. For example, the global continuity equation is multiplied by dy and integrated from y^+ to y^-

$$\int_{y^-}^{y^+} \frac{1}{r_w} \frac{\partial(r_w \rho u)}{\partial x} dy + \int_{y^-}^{y^+} \frac{\partial(\rho v)}{\partial y} dy = o$$

to yield

$$\left[\frac{1}{r_w} \frac{\partial(r_w \rho u)}{\partial x}\right]_{av} \Delta y + (\rho v)^+ - (\rho v)^- = 0$$

where $\Delta y = y^+ = y^-$ and use has been made of the mean value theorem to integrate the first term. When the limit as $\Delta y \to 0$ of the equation above is taken the result is

$$(\rho v)^- = (\rho v)^+ \qquad (2.31)$$

This same procedure was used with the other equations and the resulting relations are given in Table 2.6.

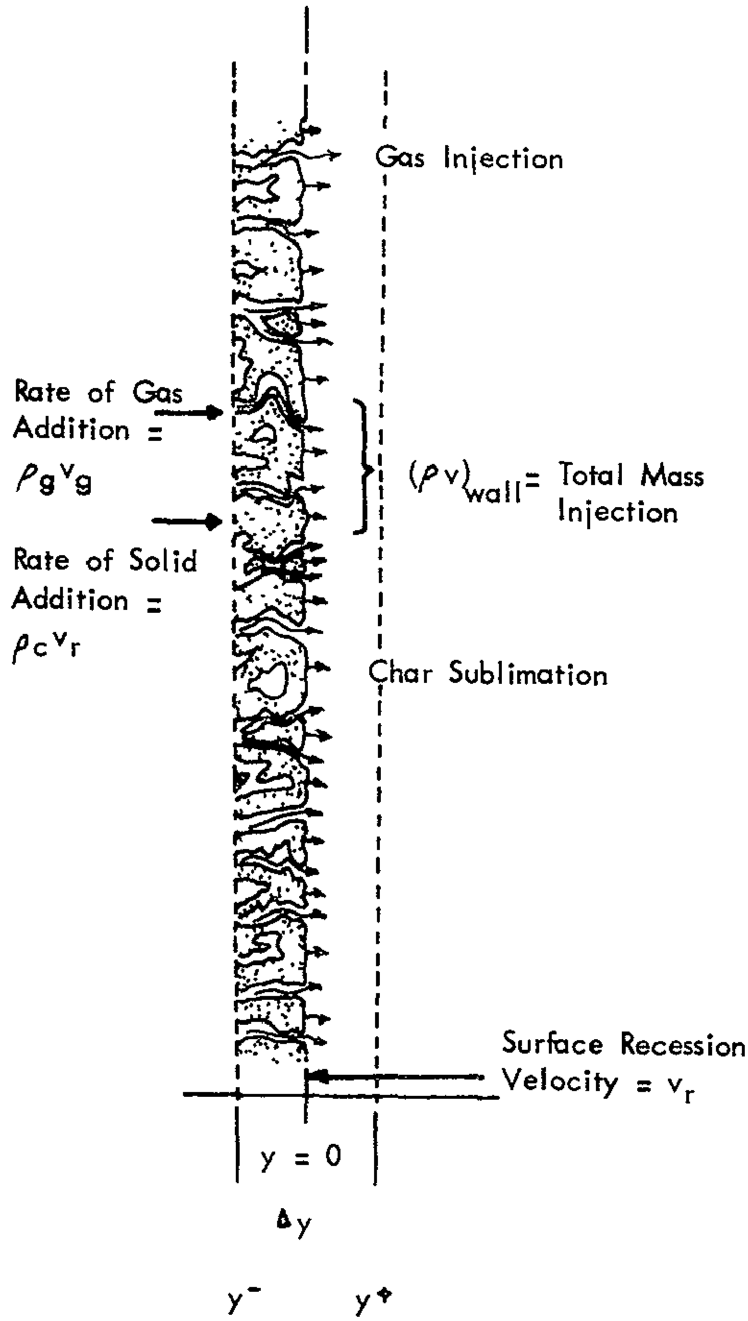


Figure 2.2 Illustration of Overall Surface Material Balance

TABLE 2.6

SURFACE BALANCE EQUATIONS

Global Mass Balance:

$$(\rho v)^{-} = (\rho v)^{+} \tag{2.31}$$

Species Mass Balance:

$$(\rho v c_i - J_i)^{-} = (\rho v c_i - J_i)^{+} \tag{2.32}$$

X-momentum:

$$(\rho v u - \mu \frac{\partial u}{\partial y})^{-} = (\rho v u - \mu \frac{\partial u}{\partial y})^{+} \tag{2.33}$$

y-momentum:

$$P^{-} = P^{+} \tag{2.34}$$

Energy:

$$(\rho v H - k \frac{\partial T}{\partial y} + \sum_i h_i J_{i,y} + q_{R,y} - \mu u \frac{\partial u}{\partial y})^{-} \tag{2.35}$$

$$= (\rho v H - k \frac{\partial T}{\partial y} + \sum_i h_i J_{i,y} + q_{R,y} - \mu u \frac{\partial u}{\partial y})^{+}$$

In these surface balance equations all the dependent variables on the char side of the interface (those with a superscript –) must be interpreted as being dependent upon both the solid and gaseous phases. For example $\rho^- = \rho_g + \rho_c$ and $(\rho v)^- = \rho_g v_g + \rho_c v_c$. In addition it must be noticed that if the flow in the char is one dimensional ($u^- = 0$), as it is usually assummed to be, the x – momentum equation is identically zero while the terms with arrows through them in the energy balance also become zero. This same situation exists on the stagnation line since u is zero for all values of y.

Shock Boundary Conditions

The bow shock is governed by three basic conservation equations, corresponding to the three physical principles of conservation of mass, momentum, and energy. These equations, commonly known as the Rankine-Hugoniot equations are given in rectangular coordinates by (Ref. 2.6):

Continuity

$$\rho_\infty^* \; V_{\infty,n}^* = \rho_\delta^* \; V_{\delta,n}^* \tag{2.36}$$

Momentum

(normal)
$$p_\infty^* + \rho_\infty^* \; V_{\infty,n}^{*2} = p_\delta^* + \rho_\delta^* \; V_{\delta,n}^{*2} \tag{2.37}$$

(tangential)
$$V_{\infty,t}^* = V_{\delta,t}^* \tag{2.38}$$

Energy

$$h_\infty^* + \frac{1}{2} \; V_{\infty,n}^{*2} + \frac{1}{2} V_{\infty,t}^{*2} = h_\delta^* + \frac{1}{2} V_{\delta,n}^{*2} + \frac{1}{2} V_{\delta,t}^{*2} \tag{2.39}$$

Using Figure 2.3 the above equations can be written in body oriented coordinates. The development of these equations in curvilinear coordinates was performed by Engel (Ref. 2.1) and the resulting equations are given below in dimensionless form

$$v_\delta = \sin\phi \sin e - \bar{\rho} \cos\phi \cos\varepsilon \tag{2.40}$$

$$u_\delta = \sin\phi \cos e - \bar{\rho} \cos\phi \sin\varepsilon \tag{2.41}$$

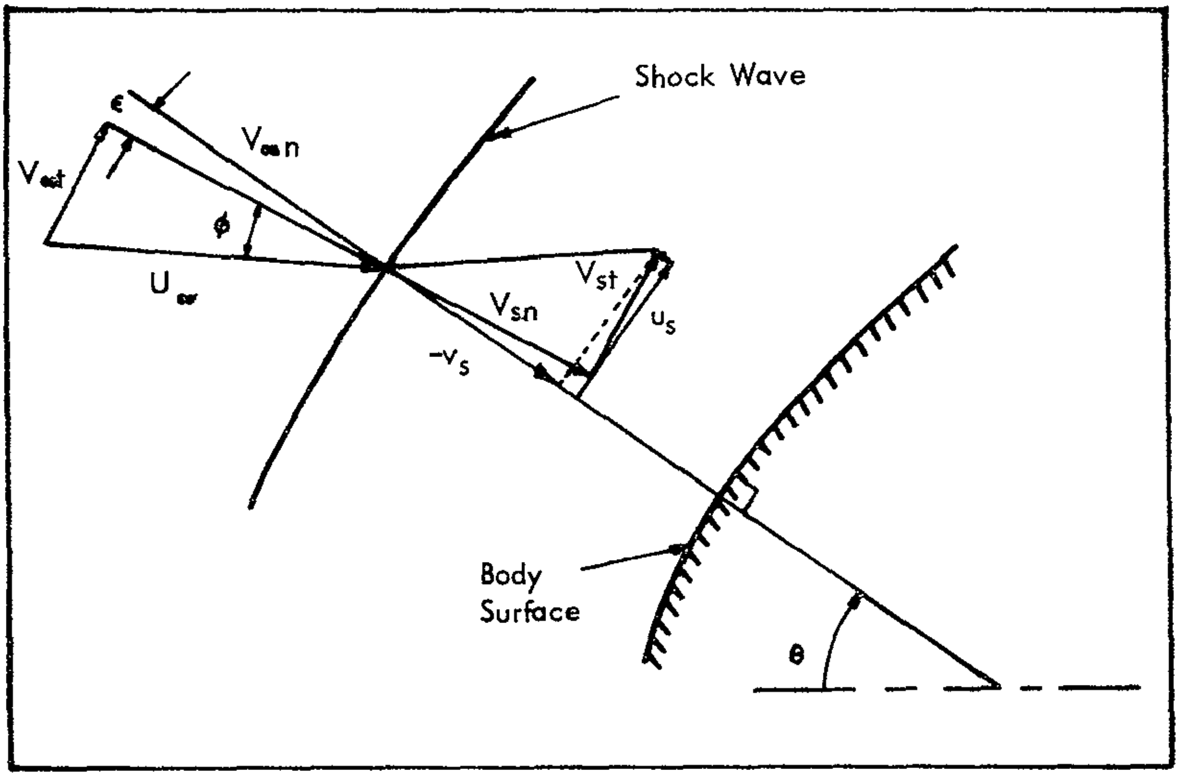


Figure 2.3 Resolution of Velocity Components in a Body-Oriented Coordinate System

$$P_\delta = (1-\bar{\rho}) \cos^2 \phi + \cancel{P_\infty}^{\,0} \tag{2.42}$$

$$h_\delta = (1-\bar{\rho}^{-2}) \cos \phi + \cancel{h_\infty}^{\,0} \tag{2.43}$$

Again, the dimensionless variables used above were defined as given in Table 2.2. As shown in Eqs. 2.42 and 2.43 the terms P_∞ and h_∞ may be neglected since they are of order $\bar{\rho}^{-2}$.

For a given shock shape and flight conditions, if the post-shock fluid is assumed to be pure air in chemical equilibrium, i.e., no diffusion of ablation products to the shock, then the velocity of the fluid at the shock and also its thermodynamic state (pressure, enthalpy, composition, etc.) can be found from Eqs. 2.40 - 2.43 and a technique for obtaining the equilibrium composition of a fluid. In practice a chemical equilibrium program is usually available and a value of $\bar{\rho}$ is guessed, from Eqs. 2.42 and 2.43 P_δ and h_δ are computed and the equilibrium program is used to obtain the composition and a new value of $\bar{\rho}$ is obtained. This procedure is repeated until the guessed and computed values of $\bar{\rho}$ agree.

Therefore, boundary conditions of the first kind, i.e., the dependent variables are specified, may be used at the shock ($y = \delta$). These boundary conditions are given below

$$\begin{aligned} u &= u_\delta \\ v &= v_\delta \\ P &= P_\delta \\ h &= h_\delta \\ Ci &= Ci_\delta\,(P\ ,\ h) \text{ (Assuming chemical equilibrium} \\ I_{\nu}.(T_{w\delta}\ ) &= 0 \text{ (No precursor radiation)} \end{aligned} \tag{2.44}$$

The additional boundary condition shown above results from assuming that no radiation from the free-stream air crosses the shock.

STAGNATION LINE BOUNDARY LAYER EQUATIONS*

The stagnation line boundary layer equations are obtained by formally taking the limit as $x \to 0$ of the boundary layer equations (Eqs. 2.26 - 2.30). The resulting equations were obtained by Engel (Ref. 2.1) and are given here in dimensional form for axisymmetric flow in Table 2.7. Some comments concerning the derivation of the equations in Table 2.7 are pertinent: 1) at the stagnation line the component of velocity tangential to the body (u) is zero because of symmetry; 2) the stagnation line x-momentum equation is obtained by first differentiating the x-momentum equation (Eq. 2.28) with respect to x and then taking the limit as $x \to 0$ of the resulting equation. This procedure is used because taking the limit of the x-momentum equation yields $(\partial P/\partial x)_{x = 0} = 0$, a relationship which does not contain any useful information.

The wall and shock boundary conditions needed in order to be able to integrate Eqs. 2.45 - 2.49 were obtained by taking the limit of Eqs. 2.31 - 2.35 and 2.40 - 2.43 as $x \to 0$. Before taking the limit of the surface balance equations the flow in the char was assumed to be one-dimensional ($u^- = u^+ = 0$). As was discussed earlier, this condition implies that the terms crossed by an arrow in the energy balance equation are zero while the x-momentum balance is identically zero. In lieu of the x-momentum balance, the fact that u is zero for all x locations at the wall ($(\partial u/\partial x)^+ = 0$) may be used with the stagnation line global continuity equation (Eq. 2.45) to yield

$$\frac{\partial (\rho' v)^+}{\partial y} = 0$$

*In this section all variables are dimensional.

TABLE 2.7

STAGNATION LINE BOUNDARY LAYER EQUATIONS

Global continuity:

$$2\,\frac{\partial u}{\partial x} = -\,\frac{1}{\rho}\,\frac{\partial}{\partial y}\,(\rho v) \tag{2.45}$$

Species continuity:

$$\rho v\,\frac{\partial C_i}{\partial y} = -\,\frac{\partial}{\partial y}\,(J_{i,y}) + \omega_i \tag{2.46}$$

x - Momentum:

$$\frac{\partial}{\partial y}\left[\mu\,\frac{\partial}{\partial y}\left(\frac{1}{\rho}\,\frac{\partial}{\partial y}\,(\rho v)\right)\right] - \rho v\,\frac{\partial}{\partial y}\left(\frac{1}{\rho}\,\frac{\partial}{\partial y}\,(\rho v)\right)$$

$$+\,\frac{\rho}{2}\left(\frac{1}{\rho}\,\frac{\partial}{\partial y}\,(\rho v)\right)^2 + 2\left(\frac{\partial^2 P}{\partial x^2}\right)_{x=0} = 0 \tag{2.47}$$

y - Momentum:

$$\frac{\partial P}{\partial y} = 0 \tag{2.48}$$

Energy:

$$\rho v\,\frac{\partial H}{\partial y} = -\,\frac{\partial}{\partial y}\left[-\,k\,\frac{\partial T}{\partial y} + \sum_i h_i J_{i,y}\right.$$

$$\left. -\,\frac{\partial q_{R,y}}{\partial y}\right] \tag{2.49}$$

The stagnation line wall boundary conditions are given in Table 2.8. At the shock $\phi = 0$ and the shock boundary conditions simplify to those given in Table 2.8. The reader should notice that while all the shock boundary conditions are of the first kind, the wall boundary conditions are of the first, second and third kind.

MATHEMATICAL DESCRIPTION OF THE PROBLEM

Before any attempt is made to describe the mathematics of the problem, it is convenient to introduce into the equations the binary diffusion assumption

$$J_{i,y} = - \rho D \frac{\partial C_i}{\partial y} \qquad (i = 1,\ldots,n) \tag{2.51}$$

and use the relation

$$H = h + v^2/2 \tag{2.52}$$

to write the energy equation (Eq. 2.49) in terms of static enthalpy instead of total enthalpy. When this is done the resulting conservation equations are:

Global continuity:

$$2 \left(\frac{\partial u}{\partial x}\right)_{x=0} = - \frac{1}{\rho} \frac{d(\rho v)}{dy} \tag{2.45}$$

Species continuity:

$$\frac{d\left(\rho D \frac{dci}{dy}\right)}{dy} - \rho v \frac{dCi}{dy} + \omega_i = 0 \qquad (i = 1,\ldots,n) \tag{2.53}$$

x - momentum:

$$\frac{d\left[\mu \frac{d\left(\frac{1}{\rho} \frac{d(\rho v)}{dy}\right)}{dy}\right]}{dy} - \rho v \frac{d\left(\frac{1}{\rho} \frac{d(\rho v)}{dy}\right)}{dy} \tag{2.47}$$

$$+ \frac{\rho}{2} \left(\frac{1}{\rho} \frac{d(\rho v)}{dy}\right)^2 + \left(\frac{\partial^2 p}{\partial x^2}\right)_{x = 0} = 0$$

TABLE 2.8

WALL AND SHOCK BOUNDARY CONDITIONS FOR STAGNATION LINE PROBLEM

Wall $(y = 0)$:

$$\rho v = (\rho v)^- \quad (2.50a)$$

$$\rho v C_i - J_i = (\rho v C_i - J_i)^- \quad (2.50b)$$

$$\frac{\partial(\rho v)}{\partial y} = 0 \quad (2.50c)$$

$$P = P^- \quad (2.50d)$$

$$\left(\rho v H - k\frac{\partial T}{\partial y} + \sum_i h_i J_{i,y} + q_{R,y}\right) = \left(\rho v H - k\frac{\partial T}{\partial y} + \sum_i h_i J_{i,y} + q_{R,y}\right)^- \quad (2.50e)$$

Shock $(y = \delta)$:

$$v = v_\delta = -\frac{\rho_\infty}{\delta,o} U_\infty \operatorname{Cos} \varepsilon \quad (2.50f)$$

$$u = u_\delta \frac{P_\infty U_{\delta,o}}{P_{\delta,o}} \sin \varepsilon \quad (2.50g)$$

$$P = P_\delta = \left(1 - \frac{\rho_\infty}{\rho_{\delta,o}}\right) \rho_\infty U_\infty \quad (2.50h)$$

$$h = h_\delta = \left(1 - \left(\frac{\rho_\infty}{\rho_{\delta,o}}\right)^2\right) \frac{U_\infty}{2} \quad (2.50i)$$

$$C_i = C_i (\rho_{\delta,o}, h_\delta) \quad (2.50j)$$

y - momentum:

$$\frac{dP}{dy} = 0 \tag{2.48}$$

Energy:

$$\rho v \frac{dh}{dy} + \rho v^2 \frac{dv}{dy} = \frac{d}{dy}\left(k \frac{dT}{dy} - \rho D \sum_i^n h_i \frac{dC_i}{dy}\right)$$

$$- \frac{dq_{r,y}}{dy} \tag{2.54}$$

The mathematical model of the stagnation line is given by the above set of 4 + n (for a system with n chemical species) ordinary integro-differential equations and a set of two-point boundary conditions given in Table 2.8. The integro-differential equations are coupled, nonlinear, of first, second and third order. They are integral since the flux divergence term ($dq_{r,y}/dy$) in the energy equation is an integral given by Eq. 2.14. These equations contain the 7 + n unknowns ρ,v, $(\partial u/\partial x)_{x=0}$, P, h, T, and C_i (i = 1,...,n) where both $(\partial u/\partial x)_{x=0}$ and $(\partial^2 P/\partial x^2)_{x=0}$ are at most functions of y. This discussion assumes that thermodynamic, transport, radiative, and kinetics properties (D, ω_i, μ, k, and h_i) are known in terms of the dependent variables listed above. These properties will be discussed in Chapter 3. It is evident that since the 4 + n equations contain 7 + n unknowns three more independent relations are needed. These are provided by the thermal equation of state

$$P = \left(\sum_i^n \frac{C_i}{M_i}\right) \rho RT \tag{2.55}$$

the caloric equation of state;

$$h = \sum_i^n C_i h_i \tag{2.56}$$

and the Rankine-Hugoniot relation for the shock pressure as a function of the shock angle measured from the stagnation line

$$P_\delta = (1 - \bar{\rho}) \cos^2 \phi \rho_\infty \quad U_\infty{}^2 \tag{2.42}$$

From the y-momentum equation and the fact that $(\frac{\partial P}{\partial x})_{x = 0} = 0$ (as discussed in the previous section) it is evident that

$$(\frac{\partial^2 P}{\partial x^2})_{x = 0} = (\frac{\partial^2 P\delta}{\partial x^2})_{x = 0} = \text{constant} \tag{2.57}$$

Substitution of Eq. 2.42 in Eq. 2.57 gives

$$(\frac{\partial^2 P}{\partial x^2})_{x = 0} = -2 \ (1 - \bar{\rho}) \ (\frac{d\phi}{dx})^2_{x = 0} \rho_\infty \quad U_\infty{}^2 \tag{2.58}$$

This development makes it possible to rewrite the x-momentum equation as x-momentum

$$\frac{d[\mu \frac{d(\frac{1}{\rho} \frac{d(\rho v)}{dy})}{dy}]}{dy} - \rho v \frac{d(\frac{1}{\rho} \frac{d(\rho v)}{dy})}{dy} \tag{2.59}$$

$$+ \frac{\rho}{2} (\frac{1}{\rho} \frac{d(\rho v)}{dy})^2 - 4 \ (1 - \bar{\rho}) \ (\frac{d\phi}{dx})^2_{x = 0} \ \rho_\infty \ U_\infty{}^2$$

Therefore, for a known shock geometry (known ϕ (x)) the conservation equations contain 6 + n unknowns instead of 7 + n as before. A further reduction in the number of equations that need to be solved may be accomplished by eliminating h from the energy equation by substituting the caloric equation of state as follows. From Eq. 2.56

$$\frac{dh}{dy} = \sum_i^n h_i \frac{dC_i}{dy} + \sum_i^n C_i \frac{dh_i}{dy}$$

$$= \sum_i^n h_i \frac{dC_i}{dy} + (\sum_i^n C_i \frac{dh_i}{dT}) \frac{dT}{dy} \tag{2.60}$$

$$= \sum_i^n h_i \frac{dC_i}{dy} + (\sum_i^n C_i C_{Pi}) \frac{dT}{dy}$$

$$= \sum_i^n h_i \frac{dC_i}{dy} + C_P \frac{dT}{dy}$$

Substituting Eq. 2.60 in Eq. 2.54 gives

Energy:

$$\rho v \, C_P \frac{dT}{dy} = - \rho v^2 \frac{dv}{dy} - \rho v \, (\sum_i^n h_i \frac{dC_i}{dy})$$

$$+ \frac{d[k \frac{dT}{dy} + \rho D (\sum_i^n h_i \frac{dC_i}{dy})]}{dy} - \frac{dq_{R,y}}{dy} \tag{2.61}$$

In this manner the unknowns have been reduced to ρ, v, $(\partial u/\partial x)_{x = 0}$, P, T, and C_i ($i = 1,\ldots,n$).

At this point it is pertinent to ask, how many of Eqs. 2.45, 2.53, 2.48, 2.55, 2.59, and 2.61 must be solved simultaneously? It is evident that the y-momentum equation (Eq. 2.48), containing only one dependent variable, is not coupled to the other equations. In fact, it can be integrated anatically to yield

$$P(y) = P_\delta = \text{constant} \tag{2.62}$$

In addition, the global continuity equation (Eq. 2.45) contains only two dependent variables and can be left out of the set that must be solved initially. This has the effect of eliminating $(\partial u/\partial x)_{x = 0}$ from the unknowns in the equations that must be solved. This variable may be readily obtained from Eq. 2.45 once the ρ and v profiles are known. The answer to the question that was asked at the beginning of this paragraph is that Eqs. 2.53, 2.55, and 2.59, and 2.61 in the unknowns ρ, v, T, and C_i ($i = 1,\ldots, n$) must be solved simultaneously.

Let us now consider the boundary conditions that must be used with these equations. The species continuity equation (Eq. 2.53) being second order requires two boundary conditions for C_i. These may be provided by Eqs. 2.50 b and 2.50 j

$$\rho v C_i + \frac{dC_i}{dy} = (\rho v C_i + \rho D \frac{dC_i}{dy})^- \quad \text{at } y = 0 \tag{2.63}$$

$$C_i = C_{i,\delta} \qquad \text{at } y = \delta \tag{2.50j}$$

Notice that the binary diffusion assumption (Eq. 2.51) was substituted in Eq. 2.50b to obtain Eq. 2.63. The x-momentum equation (Eq. 2.59) is third order and requires three boundary conditions for ρv. These are given by Eqs. 2.50a, 2.50c,

$$\rho v = (\rho v)w \text{ at } y = 0 \tag{2.50a}$$

$$\frac{d(\rho v)}{dy} = 0 \quad \text{at } y = 0 \tag{2.50c}$$

and

$$\rho v = \rho_\delta v_\delta \quad \text{at } y = \delta \tag{2.64}$$

The energy equation (Eq. 2.61) being second order in T requires two boundary conditions in addition to the ones already discussed in connection with radiation. The first of these boundary conditions is provided by the wall energy balance (Eq. 2.50e), this relation is written in terms of the variables previously discussed by substituting Eqs. 2.51, 2.52 and 2.56

$$\rho v(\sum_i^n Cihi + \frac{v^2}{2}) - k\frac{dT}{dy} - \rho D (\sum_i^n hi \frac{dC_i}{dy}) + q_{R,y} =$$

$$[\rho v(\sum_i^n Cihi + \frac{v^2}{2}) - k\frac{dT}{dy} - \rho D (\sum_i^n hi \frac{dC_i}{dy}) + q_{R,y}]^- \tag{2.65}$$

The second temperature boundary condition needed is given by

$$T = T_\delta \quad \text{at } y = \delta \tag{2.65}$$

This completes the boundary conditions needed. Of these the most complex ones are those given by Eqs. 2.63 and 2.65. The boundary condition given by Eq. 2.63 may be simplified by noticing that for fairly high mass injection rates ($(\rho v)_{wall}/\rho_\infty U_\infty > 0.05$) mass transfer by diffusion is small compared to mass transfer by convection near the wall. This means that the diffusion terms in Eq. 2.63 may be neglected to yield

$$Ci = Ci^- \quad \text{at } y = 0 \ (i = 1,\ldots n) \tag{2.67}$$

where use has been made of Eq. 2.50a to cancel the ρv terms from the equation above. The wall energy balance (Eq. 2.65) may also be simplified by neglecting the kinetic energy of the fluid ($\frac{v^2}{2}$) since it is small when compared to the thermal energy. When this is done Eq. 2.65 becomes

$$\rho v\left(\sum_i^n Cihi\right) - k\frac{dT}{dy} - \rho D\left(\sum_i^n hi\frac{dC_i}{dy}\right) + q_{R,y} = \left[\rho v\left(\sum_i^n Cihi\right) - k\frac{dT}{dy} - \rho D\left(\sum_i^n hi\frac{dC_i}{dy}\right) + q_{R,y}\right]^- \tag{2.68}$$

Since all the other boundary conditions are of the are of the first and second kind and uncoupled, while Eq. 2.68 is of the third kind and coupled to the species continuity equation, it was expedient to assume that the char is at the sublimation temperature of carbon:

$$T = Tw \quad \text{at } y = 0 \tag{2.69}$$

Until now we have assumed that δ (the shock stand-off distance) is known, a necessary condition to be able to integrate the equations of the flow-field. In practice, δ is unknown and therefore must be guessed to be able to compute it correctly. This boundary condition is obtained by

evaluating the global continuity equation at the shock

$$\frac{d(\rho v)}{dy} = -2\rho_{s,o}\left(\frac{\partial u_\delta}{\partial x}\right)_{x=0} \quad \text{at } y = \delta \tag{2.70}$$

Table 2.9 gives a summary of the stagnation line model including the conservation equations, the equation of state and the boundary conditions.

REFERENCES

2.1 Engel, Carl D., Ablation and Radiation Coupled Viscous Hypersonic Shock Layers, Ph.D. Dissertation, Louisiana State University, Baton Rouge, Louisiana (1971)

2.2 Hayes, W.D. and Probstein, R.F., Hypersonic Flow Theory, 2nd Ed., Ch. 5, Academic Press, New York (1966)

2.3 Schlichting, H., Boundary Layer Theory, 6th Ed., McGraw-Hill Co., Inc., New York (1968).

2.4 U.S. Standard Atmosphere, 1962, U.S. Government Printing Office, December, 1962

2.5 Hansen, C.F., "Approximations for the Thermodynamic and Transport Properties of High-Temperature Air," NASA Technical Report R-50, 1959

2.6 Spradley, L.W., and Engel, C.D., "Formulation of a Method for Predicting Coupled Convective and Radiative Heat Transfer About a Blunt Body," NASA CR 61200 (April, 1968)

TABLE 2.9

SUMMARY OF THE STAGNATION LINE MODEL

Governing Equations:

Species continuity:

$$\frac{d(\rho D \frac{dC_i}{dy})}{dy} - \rho v \frac{dC_i}{dy} + \omega_i = 0 \tag{2.53}$$

X-momentum:

$$\frac{d[\mu \frac{d(\frac{1}{\rho} \frac{d(\rho v)}{dy})}{dy}]}{dy} - \rho v \frac{d(\frac{1}{\rho} \frac{d(\rho v)}{dy})}{dy} \tag{2.59}$$

$$+ \frac{\rho}{2} \left(\frac{1}{\rho} \frac{d(\rho v)}{dy}\right)^2 - 4 \; (1-\bar{\rho}) \; \left(\frac{\partial u}{\partial x}\right)^2_{x = 0} \rho_\infty \, U_\infty^{\;2} = 0$$

Energy:

$$\frac{d(k \frac{dT}{dy})}{dy} - \rho v C_p \frac{dT}{dy} + \frac{d[\rho D \; (\sum_i^n h_i \frac{dC_i}{dy})]}{dy}$$

$$- \rho v^2 \frac{dv}{dy} - \rho v . (\sum_i^n h_i \frac{dC_i}{dy}) - \frac{dq_{R,y}}{dy} = 0 \tag{2.61}$$

Thermal equation of state:

$$\rho \; 1R \; T \; (\sum_i^n \frac{C_i}{M_i}) = P_\delta \tag{2.55}$$

TABLE 2.9 (CONTINUED)

Boundary Conditions:

Wall ($y = 0$)		Shock ($y = \delta$)	
1. $\rho v = (\rho v)_w$	(2.50a)	1. $\rho v = (\rho v)_\delta$	(2.64)
2. $\frac{d(\rho v)}{dy} = 0$	(2.50c)	2. $\frac{d(\rho v)}{dy} = -2\rho_{\delta,o} \left(\frac{\partial u_\delta}{\partial x}\right)_{x=0}$	(2.70)
3. $C_i = C_{i,w}$	(2.67)	3. $C_i = C_{i,\delta}$	(2.50j)
4. $T = T_w$	(2.68)	4. $T = T_\delta$	(2.69)

CHAPTER 3

THERMODYNAMIC, TRANSPORT, KINETIC, AND RADIATIVE PROPERTIES

The thermodynamic, transport, kinetic, and radiative properties of the mixture of air and ablation species which are necessary to solve the conservation equations that were presented in the previous chapter are evaluated with the methods described in this chapter.

THERMODYNAMIC PROPERTIES

Thermodynamic Properties of Pure Species at Standard State

The specific heat at constant pressure, enthalpy, entropy, and Gibbs free energy of the pure species at one atmosphere pressure (standard state) were obtained by Esch, et. al. (Ref. 3.1). This was accomplished by curve fitting with the method of least squares, temperature polynomials to data found in the literature. The polynomials used to fit the thermodynamic properties are given in Table 3.1. For each species, two different fits were made, one for low temperatures ($1000^{\circ}K \leq T \leq 6000^{\circ}K$) and the other for high temperatures ($6000^{\circ}K \leq T \leq 15,000^{\circ}K$). For each species, the coefficients for the expressions in Table 3.1 are given in Table 3.2 for the two temperature ranges. Each term in Eqs. 3.1 - 3.4 is dimensionless, temperature is in $^{\circ}K$ and $\mathbb{R} = 1.987$ cal/gmole - $^{\circ}K$, therefore the units of the thermodynamic properties are as given below

$$C_{p_i} = [\text{ cal/gmole of i} - {}^{\circ}K]$$

TABLE 3.1 POLYNOMINAL EQUATIONS FOR STANDARD THERMODYNAMIC PROPERTIES

Specific Heat

$$\frac{C^{o}_{p,i}}{R} = A_{1,i} + A_{2,i}T + A_{3,i}T^{2} + A_{4,i}T^{3} + A_{5,i}T^{4} \qquad (3.1)$$

Enthalpy

$$\frac{h^{o}_{i}}{RT} = A_{1,i} + \frac{A_{2,i}}{2}T + \frac{A_{3,i}}{3}T^{2} + \frac{A_{4,i}}{4}T^{3} + \frac{A_{5,i}}{5}T^{4} + \frac{A_{6,i}}{T} \qquad (3.2)$$

Entropy

$$\frac{S^{o}_{i}}{R} = A_{1,i}\ln T + A_{2,i}T + \frac{A_{3,i}}{2}T^{2} + \frac{A_{4,i}}{3}T^{3} + \frac{A_{5,i}}{4}T^{4} + A_{7,i} \qquad (3.3)$$

Free Energy

$$\frac{F^{o}_{i}}{RT} = A_{1,i}(1-\ln T) - \frac{A_{2,i}}{2}T - \frac{A_{3,i}}{6}T^{2} - \frac{A_{4,i}}{12}T^{3} - \frac{A_{5,i}}{20}T^{4} + \frac{A_{6,i}}{T} - A_{7,i} \qquad (3.4)$$

TABLE 3.2

POLYNOMIAL COEFFICIENTS FOR THERMODYNAMIC PROPERTY CORRELATIONS

Species	$A_{1,i}$	$A_{2,i}$	$A_{3,i}$	$A_{4,i}$	$A_{5,i}$	$A_{6,i}$	$A_{7,i}$	T*
C+	0.2609E 01	-0.1393E-03	0.5959E-07	-0.1037E-10	0.6345E-15	0.2168E 06	0.3709E 01	L
	0.2528E 01	0.4869E-05	-0.7026E-08	0.1134E-11	-0.3476E-16	0.2168E 06	0.4139E 01	H
H	0.2500E 01	-0.8243E-06	0.6421E-09	-0.1720E-12	0.1457E-16	0.2547E 05	-0.4612E 00	L
	0.3934E 01	-0.1776E-02	0.6013E-06	-0.7819E-10	0.3482E-14	0.2547E 05	-0.8598E 01	H
N+	0.2727E 01	-0.2820E-03	0.1105E-06	-0.1551E-10	0.7847E-15	0.2254E 06	0.3645E 01	L
	0.2499E 01	-0.3725E-05	0.1147E-07	-0.1102E-11	0.3078E-16	0.2254E 06	0.4950E 01	H
O+	0.2491E 01	0.2762E-04	-0.1881E-07	0.3807E-11	-0.1028E-15	0.1879E 06	0.4424E 01	L
	0.2944E 01	-0.4108E-03	0.9156E-07	-0.5848E-11	0.1190E-15	0.1879E 06	0.1750E 01	H
E-	0.2500E 01	0.3440E-06	-0.1954E-09	0.3937E-13	-0.2573E-17	-0.7450E 03	-0.1173E 02	L
	0.2508E 01	-0.6332E-05	0.1364E-08	-0.1094E-12	0.2934E-17	-0.7450E 03	-0.1208E 02	H
C	0.2612E 01	-0.2030E-03	0.1095E-06	-0.1695E-10	0.8590E-15	0.8542E 05	0.4144E 01	L
	0.2141E 01	0.3219E-03	-0.5498E-07	0.3604E-11	-0.5564E-16	0.8542E 05	0.6874E 01	H
CN	0.3411E 01	0.4897E-03	0.1005E-06	-0.3473E-10	0.2361E-14	0.4745E 05	0.4746E 01	L
	0.3473E 01	0.7337E-03	-0.9088E-07	0.4847E-11	-0.1018E-15	0.5420E 05	0.4152E 01	H
CO	0.3254E 01	0.9698E-03	-0.2647E-06	0.3037E-10	-0.1177E-14	-0.1434E 05	0.4875E 01	L
	0.3366E 01	0.8027E-03	-0.1968E-06	0.1940E-10	-0.5549E-15	-0.1434E 05	0.4263E 01	H
C2	0.4443E 01	-0.2885E-03	0.3036E-06	-0.6244E-10	0.3915E-14	0.9787E 05	-0.1090E 01	L
	0.4026E 01	0.4857E-03	-0.7026E-07	0.4666E-11	-0.1142E-15	0.9787E 05	0.1090E 01	H
C2H	0.3485E 01	0.3563E-02	-0.1237E-05	0.1866E-09	-0.1013E-13	0.5809E 05	0.4784E 01	L
	0.5307E 01	0.8966E-03	-0.1378E-06	0.9251E-11	-0.2278E-15	0.5809E 05	-0.5288E 01	H

* Temperature Range L = 1000-6000°K, H = 6000-15000°K

TABLE 3.2 (Continued)

Species	$A_{1,i}$	$A_{2,i}$	$A_{3,i}$	$A_{4,i}$	$A_{5,i}$	$A_{6,i}$	$A_{7,i}$	T*
C2H2	0.3891E 01	0.5717E-02	-0.1957E-05	0.2931E-09	-0.1585E-13	0.2590E 05	0.6520E 00	L
	0.6789E 01	0.1503E-02	-0.2295E-06	0.1534E-10	-0.3763E-15	0.2590E 05	-0.1539E 02	H
C3	0.4002E 01	0.3541E-02	-0.1318E-05	0.2064E-09	-0.1144E-13	0.9423E 05	0.2020E 01	L
	0.2213E 02	-0.1759E-01	0.5565E-05	-0.6758E-09	0.2825E-13	0.9423E 05	-0.1021E 03	H
C3H	0.3965E 01	0.6200E-02	-0.2265E-05	0.3717E-09	-0.2262E-13	0.6283E 05	0.3467E 01	L
	0.3965E 01	0.6200E-02	-0.2265E-05	0.3717E-09	-0.2262E-13	0.6283E 05	0.3467E 01	H
C4H	0.5874E 01	0.7403E-02	-0.2729E-05	0.4437E-09	-0.2637E-13	0.7605E 05	-0.4010E 01	L
	0.5874E 01	0.7403E-02	-0.2729E-05	0.4437E-09	-0.2637E-13	0.7605E 05	-0.4010E 01	H
HCN	0.3654E 01	0.3444E-02	-0.1258E-05	0.2169E-09	-0.1430E-13	0.1442E 05	0.2373E 01	L
	0.3654E 01	0.3444E-02	-0.1258E-05	0.2169E-09	-0.1430E-13	0.1442E 05	0.2373E 01	H
H2	0.3358E 01	0.2794E-03	0.9372E-07	-0.2948E-10	0.2141E-14	-0.1018E 04	-0.3548E 01	L
	0.3363E 01	0.4656E-03	-0.5127E-07	0.2802E-11	-0.4905E-16	-0.1018E 04	-0.3716E 01	H
N	0.2474E 01	0.9097E-04	-0.7814E-07	0.2218E-10	-0.1489E-14	0.5609E 05	0.4300E 01	L
	0.2746E 01	-0.3909E-03	0.1338E-06	-0.1191E-10	0.3369E-15	0.5609E 05	0.2872E 01	H
O	0.2670E 01	-0.1970E-03	0.7193E-07	-0.8901E-11	0.4002E-15	0.2915E 05	0.4504E 01	L
	0.2548E 01	-0.5952E-04	0.2701E-07	-0.2798E-11	0.9380E-16	0.2915E 05	0.5049E 01	H
N2	0.3221E 01	0.9878E-03	-0.2907E-06	0.3938E-10	-0.2000E-14	-0.1043E 04	0.4326E 01	L
	0.3727E 01	0.4684E-03	-0.1140E-06	0.1154E-10	-0.3293E-15	-0.1043E 04	0.1294E 01	H
O2	0.3316E 01	0.1151E-02	-0.3726E-06	0.6186E-10	-0.3666E-14	-0.1044E 04	0.5393E 01	L
	0.3721E 01	0.4254E-03	-0.2835E-07	0.6050E-12	-0.5186E-17	-0.1044E 04	0.3254E 01	H

*Temperature Range L = 1000-6000°K, H = 6000-15,000°K

$$h_i^\circ = [\text{ cal/gmole of i}]$$

$$S_i^\circ = [\text{cal/gmole of i} - {}^\circ K)$$

$$F_i^\circ = [\text{cal/gmole of i}]$$

Thermodynamic Properties of Pure Species at Arbitrary State

Expressions for the thermodynamic properties of pure species at arbitrary pressures and temperatures are predicted in the following manner. The species are assumed to be ideal gases so that (Ref. 3.2)

$$h_i = h_i\ (T) = h_i{}^\circ \tag{3.5}$$

and since $C_{Pi} = (\partial h_i/\partial T)_P$ then

$$C_{Pi} = C_{Pi}\ (T) = C_{Pi}{}^\circ \tag{3.6}$$

In other words, both the enthalpy and the specific heat at constant pressure are independent of pressure. On the other hand let us examine entropy, from the first law of thermodynamics

$$dh_i = TdS_i + \frac{1}{X_i}\ dP \tag{3.7}$$

where Xi is the molar concentration of species i (moles of i/volume). Rewritting Eq. 3.7 gives

$$dS_i = \frac{1}{T}\, dhi - \frac{1}{X_i T}\ dP \tag{3.8}$$

From the thermal equation of state for an ideal gas

$$P = X_i\, \mathbb{R}T \tag{3.9}$$

and Eq. 3.8 can be rewritten as

$$dS_i = \frac{1}{T}\, dhi - \mathbb{R}\ d\ (\ln P) \tag{3.10}$$

and integrated to yield

$$S_i\ (T,P) = \int \frac{1}{T}\, dhi - \mathbb{R} \ln P + A_{8,i} \tag{3.11}$$

where $A_{8,i}$ is an integration constant. In order to evaluate $A_{8,i}$ we make use of the fact that $S_i\ (T,1) = S_i^\circ$ and obtain (by evaluating Eq. 3.11 at P = 1 atmosphere).

$$A_{8,i} = S_i^\circ - \int \frac{1}{T}\, dhi \tag{3.12}$$

Substitution of Eq. 3.12 in Eq. 3.11 yields

$$S_i = S_i\ (T,P) = S_i^\circ - \mathbb{R} \ln P \tag{3.13}$$

The Gibbs free energy is defined as

$$F_i = hi - TS_i \tag{3.14}$$

substitution of Eqs. 3.5 and 3.13 in the above equation results in

$$F_i = hi^\circ - TS_i^\circ - \mathbb{R}T \ln P \tag{3.15}$$

or

$$F_i = F_i\ (T,P) = F_i^\circ + \mathbb{R}T \ln P \tag{3.16}$$

since

$$F_i^\circ = hi^\circ - TS_i^\circ$$

Thermodynamic Properties of Mixtures

For each species in a mixture consisting of a total of n gaseous species the thermodynamic properties are given by

$$h_i = h_i\ (T) = h_i^\circ \tag{3.17}$$

$$C_{Pi} = C_{Pi}\ (T) = C_{Pi}^\circ \tag{3.18}$$

$$S_i = S_i\ (T,P) = S_i^\circ - \mathbb{R} \ln P_i \tag{3.19}$$

$$F_i = F_i\ (T,P) = F_i^\circ + \mathbb{R}T \ln P_i \tag{3.20}$$

where P_i is the partial pressure of species i (atmosphere). Mixture properties are then given by

$$h = \sum_{i}^{n} y_i \, h_i \text{ [cal./gmole of mixture]} \tag{3.21}$$

$$C_p = \sum_{i}^{n} y_i \, C_{Pi} \text{ [cal./gmole of mixture - °K]} \tag{3.22}$$

$$S = \sum_{i}^{n} y_i \, S_i \text{ [cal./gmole of mixture - °K]} \tag{3.23}$$

$$F = \sum_{i}^{n} y_i \, F_i \text{ [cal./gmole of mixture]} \tag{3.24}$$

where yi is the i^{th} species mole fraction (gmole of i/gmole of mixture) and the units of each property are given by the terms in brackets.

TRANSPORT PROPERTIES

Presently available transport properties data for air and ablation products are extremely limited because of the difficulties involved with experimental measurements at the extremely high temperatures of interest. Thus, it is necessary to rely heavily on properties obtained from theoretical predictions like the Chapman-Enskog kinetic theory (Ref. 3.3). At the lower temperatures, where ionization has not yet begun to occur, the classical first order Chapman-Enskog kinetic theory has been found to be reasonably accurate. However for higher temperatures the use of more rigorous kinetic models becomes necessary. The transport properties used in the present work were obtained by Esch (Ref. 3.4); in order to obtain simple closed form expressions for species viscosity and thermal conductivity the theoretical results reported in the literature were curve fit with polynominal expressions in temperature.

Viscosity

The species viscosity data obtained from theoretical predictions

were curve fit to a second degree polynomial in temperature

$$\mu_i = a_i + b_i T + c_i T^2 \tag{3.25}$$

In the above relation temperature is in °K, μi is in lbm of i/ft-sec and the coefficients a_i, b_i, and c_i are given in Table 3.3. For a mixture consisting of n gaseous species viscosity was predicted by the Buddenberg-Wilke model (Ref. 3.5)

$$\mu = \sum_i^n \frac{y_i \, \mu_i}{\sum_i^n yi \, \phi_{ij}} \tag{3.26}$$

where

$$\Phi_{ij} = \frac{1}{\sqrt{8}} \left(1 + \frac{M_i}{M_j}\right)^{-\frac{1}{2}} \left[1 + \left(\frac{\mu_i}{\mu_j}\right)^{\frac{1}{2}} \left(\frac{M_j}{M_i}\right)^{\frac{1}{4}}\right]^2 \tag{3.27}$$

and M_i is the i^{th} species molecular weight.

Thermal Conductivity

The species thermal conductivities were obtained from linear fits of theoretical results:

$$K_i = d_i + e_i T \tag{3.28}$$

again T is in °K, k_i is in °K, k_i is in BTU/ft - sec - °R and the coefficients d_i and e_i are given in Table 3.4. Misture thermal conductivity is compared by

$$k = \sum_i^n \frac{y_i \, k_i}{\sum_i^n yi \, \phi_{ij}} \tag{3.29}$$

where as before ϕ_{ij} is given by Eq. 3.27.

Binary Diffusion

The coefficient of binary diffusion was assumed to be a function of temperature and pressure as follows (Ref. 3.4):

$$D = \frac{8.128 \times 10^{-8} \, T^{1.659}}{P} \tag{3.30}$$

TABLE 3.3

CONSTANTS FOR VISCOSITY CORRELATION

$$\mu_i = a_i + b_i T + c_i T^2 \quad \frac{\text{lbm}}{\text{ft-sec}}$$

Species	a x 10^5	b x 10^7	c x 10^{12}	Temperature Range (°K)
O_2	1.693	0.1496	-0.2276	2,000-10,000
N_2	0.970	0.1613	-0.1916	2,000-10,000
O	1.519	0.1875	-0.2228	2,000-10,000
N	0.253	0.2206	-0.3737	2,000-10,000
O^+	0.0	0.0500	-0.1000	8,000-15,000
N^+	0.0	0.0500	-0.1000	8,000-15,000
e^-	0.0	0.0500	-0.1000	8,000-15,000
C	1.997	0.1772	-0.3378	5,000-10,000
H	0.294	0.0889	-0.0811	4,000-10,000
H_2	-0.079	0.0791	-0.0886	4,000-10,000
CO	2.404	0.1363	-0.2184	4,000- 9,000
C_3	2.019	0.1179	-0.1655	1,000- 5,000
CN	2.404	0.1363	-0.2184	4,000- 9,000
C_2H	2.404	0.1363	-0.2184	4,000- 9,000
C_2H_2	1.396	0.0842	-0.6939	1,000- 5,000
C_3H	2.019	0.1179	-0.1655	1,000- 5,000
C_4H	2.019	0.1179	-0.1655	1,000- 5,000
HCN	1.378	0.0965	-0.0948	1,000- 5,000
C_2	1.931	0.1393	-0.2575	4,000- 9,000
C^+	0.0	0.0500	-0.1000	8,000-15,000

TABLE 3.4

CONSTANTS FOR THERMAL CONDUCTIVITY CORRELATION

$k_i = a_i + b_i T$ (BTU/ft-sec-^{o}R)

Species	a x 10^5	b x 10^8	Temperature Range (^{o}K)
O_2	1.019	0.4901	2,000-10,000
N_2	0.654	0.6457	2,000-10,000
O	1.250	0.7092	2,000-10,000
N	1.281	0.8593	2,000-10,000
O^+	26.0	0.0	8,000-15,000
N^+	26.0	0.0	8,000-15,000
e^-	26.0	0.0	8,000-15,000
C	2.506	0.7479	5,000-10,000
H	2.496	5.129	4,000-10,000
H_2	3.211	5.344	4,000-10,000
CO	0.859	0.6233	1,000- 5,000
C_3	0.630	0.5804	1,000- 5,000
CN	0.859	0.6233	2,000-10,000
C_2H	1.126	0.7439	1,000- 5,000
C_2H_2	1.126	0.7439	1,000- 5,000
C_3H	0.630	0.5804	1,000- 5,000
C_4H	0.630	0.5804	1,000- 5,000
HCN	0.486	0.8714	1,000- 5,000
C_2	0.859	0.6233	1,000- 5,000
C^+	26.0	0.0	8,000-15,000

where T is in °K, P is in atmospheres, and D is in ft^2/sec.

Chemical Kinetics

Formulation of a kinetics model that describes the chemical reactions in the flow-field requires the investigation of innumerable possible reactions that can take place among the air and ablation product species that are present in the shock layer. Because of the complexity associated with kinetics models involving a large number of reactions only those reactions that are important may be considered. In other words, we seek to develop the simplest model capable of accurately describing the chemical kinetics of the flow-field.

Suppose that a number m of chemical reactions of the form

$$\sum_{i}^{n} \nu_{ij}' S_i \rightleftarrows \sum_{i}^{n} \nu_{ij}'' S_i \quad (j = 1,\ldots,m) \tag{3.31}$$

where n is the total number of species in the system, and ν_{ij}' and ν_{ij}'' are the stoichiometric coefficients of the reactants and products in the j^{th} reaction, respectively, have been chosen as a kinetics model, then the rate of generation of species i due to finite-rate chemical reactions (ω_i) is given by the Law of Mass Action (Ref. 3.6) as

$$\begin{aligned} W_i = {} & \sum_{j}^{m} (\nu_{ij} - \nu_{ij})\, f_j\, M_j \prod_{k=1}^{n} \left(\frac{\rho C_k}{M_k}\right)^{\nu_{kj}'} \\ & - \sum_{j}^{m} (\nu_{ij} - \nu_{ij})\, r_j\, M_i \prod_{k=1}^{n} \left(\frac{\rho C_k}{M_k}\right)^{\nu_{kj}''} \quad (i = 1,\ldots,n) \end{aligned} \tag{3.32}$$

where ρ is the density, C_i is the i^{th} species mass fraction, M_i is the molecular weight of species i, and f_j and r_j are the forward and backward reaction rate constants for the j^{th} reaction, respectively. The forward and backward reaction rate constants are generally taken as functions of temperature of the form (Ref. 3.7)

$$f_j = a_{fj}\, T^{b_{fj}} \exp(-e_{fj}/t) \tag{3.33}$$

$$(j = 1,\ldots,m)$$

$$r_j = a_{rj}\, T^{b_{rj}} \exp(-e_{rj}/T) \tag{3.34}$$

where $a_{fj}\, T^{b_{fj}}$ and $a_{rj}\, T^{b_{fj}}$ are the frequency factors, and e_{fj} and e_{rj} are the activation energies for the j^{th} forward and backward reactions, respectively.

In practice, the forward reaction rate constant is normally measured experimentally while the backward reaction rate constant is obtained as follows: At equilibrium

$$(\nu''_{ij} - \nu'_{ij})\, f_j\, M_i \prod_{k=1}^{n} \left(\frac{\rho C_k}{M_k}\right)^{\nu'_{kj}} = (\nu''_{ij} - \nu'_{ij})\, r_j\, M_i \prod_{k=1}^{n} \left(\frac{\rho C_k}{M_k}\right)^{\nu''_{ij}} \tag{3.35}$$

or

$$\frac{f_j}{r_j} = \prod_{k=1}^{n} \left(\frac{\rho C_k}{M_k}\right)^{(\nu''_{ki} - \nu'_{kj})} \qquad (j = 1,\ldots,m) \tag{3.36}$$

The equilibrium constant for the j^{th} reaction (K_j) is given by (Ref. 3.8)

$$K_j = (\mathbb{R}T)^{\sum_k^n (\nu''_{kj} - \nu'_{kj})} \prod_{k=1}^{n} \left(\frac{\rho C_k}{M_k}\right)^{(\nu''_{kj} - \nu'_{kj})} \qquad (j = 1,\ldots,m) \tag{3.37}$$

where $\mathbb{R}$ is the ideal gas constant. Substitution of Eq. 3.37 in Eq. 3.36 and solving for r_j yields

$$r_j = (\mathbb{R}T)^{\sum_k^n (\nu''_{kj} - \nu'_{kj})}\, f_j\, K_j^{-1} \qquad (j = 1,\ldots,m) \tag{3.38}$$

Therefore, knowing the forward reaction rate constant and the equilibrium constant the backward rate constant can be obtained from Eq. 3.38.

There are two methods commonly used to obtain the equilibrium constant of a reaction. The first method consists of computing the chemical equilibrium composition of the system for a range of different temperatures and using this data with Eq. 3.37 to compute the equilibrium constant for different values of temperature. This set of values of equilibrium constant versus temperature can be curve fit with

analytical expression for K_j

$$K_j = \Gamma_j \exp(\beta_j/T) \qquad (j = 1,\ldots,m) \tag{3.39}$$

where Γ_j and β_j are constants. The second method consists of using the expression for the equilibrium constant obtained from thermodynamics (Ref. 3.9)

$$K_j = \exp(-\Delta F_j°/1RT) \qquad (j = 1,\ldots,m) \tag{3.40}$$

where $\Delta F_j°$ is the standard free energy for the j^{th} reaction and is given by

$$\Delta F_j° = \sum_{k=1}^{n} (\nu''_{kj} - \nu'_{kj}) F° \qquad (j = 1,\ldots,m) \tag{3.41}$$

$$\frac{\Delta F_j}{RT} = B1_j (1 - \ln T) - \frac{B_{2,j}}{2} T - \frac{B_{3,j}}{6} T^2 - \frac{B_{4,j}}{12} T^3$$

$$- \frac{B_{5,j}}{20} T^4 + \frac{B_{6,j}}{T} - B_{7,j}$$

where

$$B_{\ell,j} = \sum_{k=1}^{n} (\nu''_{kj} - \nu'_{kj}) A_{\ell,k} \qquad (1 = 1,\ldots,7)$$

Therefore the equilibrium constant may be obtained from the species standard free energy expressions given in Tables 3.1 and 3.2. The first method has the advantage that when it is used the backward reaction rate constant has the same form as Eq. 3.34; this can be seen by eliminating f_j and K_j from Eq. 3.38 by substituting Eqs. 3.33 and 3.39 to yield

$$r_j = \frac{(RT)^{\sum_k^n (V''_{kj} - V'_{kj})} a_{fj} T^{b_{fj}}}{\Gamma_j} \exp[-(e_{fj} + \beta_j)/T] \tag{3.42}$$
$$(j = 1,\ldots,m)$$

Comparing Eqs. 3.42 and 3.34 it is evident that

$$a_{rj} = \frac{(R)^{\sum_k^n (\nu''_{kj} - \nu'_{kj})} a_{fj}}{\Gamma_j}$$

$$b_{rj} = b_{fj} + \sum_k^n (\nu''_{kj} - \nu'_{kj}) \tag{3.43}$$

$$e_{rj} = e_{fj} + \beta_j$$

The second method, on the other hand, does not yield an expression of

the form of Eq. 3.34 since $\Delta F_j{}^\circ$ is a polynomial expression (see Eq. 3.4). The resulting expression is therefore cumbersome to use. On the other hand the second method has the advantage that K_j is readily obtained from the species standard free energy and does not require a routine for computing chemical equilibrium as does the first method.

Up until this point, this discussion has been limited to the case where the kinetics model was already determined; however, as will be shown below, establishing an appropriate kinetics model is not a simple task. Information has been collected over a period of years on the chemistry of the reactions of ablation products and air species, and this information is stored in the form of a computer implemented data management file which presently contains several thousand reactions. It was necessary to examine this extensive set of reactions to arrive at a listing of most probable reactions (Ref. 3.10). These probable reactions were selected involving 28 species. Currently, it is not possible to consider a solution of the conservation equations with this many species and reactions. Furthermore, rate data were not available for many of them. Therefore, this preliminary list was then reexamined for key reactions to represent the chemical system. The selection was based on the species anticipated to have the largest compositions and presumably, therefore, dominate the energy level of the system. These key reactions are given in Table 3.5. In Table 3.6 the previously omitted ablation product and combustion reactions are given. Shown in Table 3.7 are the additional air and hydrogen combustion reactions. The values of the coefficients a_{fj}, b_{fj}, and e_{fj} of the forward reaction rate constants (see Eq. 3.33) for the reactions in Table 3.5 are given in Table 3.8. The forward rate constants listed

TABLE 3.5

SELECTED IMPORTANT CHEMICAL REACTIONS IN

THE SHOCK LAYER OF AN ABLATING BODY

1.	CO	+	N	=	CN	+	O		
2.	C_2H	+	H	=	C_2	+	H_2		
3.	N_2	+	M	=	2N	+	M		
4.	H_2	+	M	=	2H	+	M		
5.	O	+	M	=	O^+	+	E –	+	M
6.	N	+	M	=	N^+	+	E–	+	M
7.	CN	+	M	=	C	+	N	+	M
8.	CHN	+	M	=	CN	+	H	+	M
9.	C_2H	+	M	=	C_2	+	H	+	M
10.	C_2H_2	+	M	=	C_2H	+	H	+	M
11.	CO	+	M	=	C	+	O	+	M
12.	C_3	+	M	=	C_2	+	C	+	M
13.	C_2	+	M	=	2C	+	M		
14.	C	+	M	=	C^+	+	E–	+	M
15.	C_3H	+	M	=	C_2H	+	C	+	M
16.	C_4H	+	M	=	C_3H	+	C	+	M

TABLE 3.6 ADDITIONAL ABLATION PRODUCT AND COMBUSTION REACTIONS

<u>CH_4, CH_3, CH_2, CH Reactions</u>

1. $CH_4 = CH_3 + H$
2. $CH_4 = CH_2 + H_2$
3. $CH_3 = CH_2 + H$
4. $CH_2 = CH + H$
5. $CH_2 + H = CH + H_2$
6. $CH = C + H$
7. $C + H_2 = CH + H$
8. $CH_2 + O = CO + H_2$

<u>C_2H_2, C_2H, Reactions</u>

1. $C_2H_2 + H = C_2H + H_2$
2. $C_2H_2 + O = CH_2 + CO$
3. $C_2H_2 + OH = C_2H + H_2O$
4. $C_2H + H = C_2 + H_2$
5. $C_2H + O = CH + CO$

<u>Other Reactions</u>

1. $C_2 + H = C + CH$

<u>CN Reactions</u>

1. $2CN = C_2 + N_2$
2. $CN + O = N + CO$

TABLE 3.7 ADDITIONAL AIR AND HYDROGEN COMBUSTION REACTIONS

Air Reactions	Hydrogen Combustion Reactions
1. $O_2 + M = 2O + M$	1. $H + O_2 = OH + O$
2. $N_2 + M = N^+ + e^- + M$	2. $O + H_2 = OH + H$
3. $NO + M = N + O + M$	3. $OH + H_2 = H_2O + H$
4. $NO + O = O_2 + N$	4. $2OH = H_2O + O$
5. $N_2 + O = NO + N$	5. $H + OH + M = H_2O + M$
6. $N + O = NO^+ + e^-$	6, $H + O + M = OH + M$

TABLE 3.8 COEFFICIENTS OF THE FORWARD REACTION RATE CONSTANTS

j^{th} Reaction	a_{fj}	b_{fj}	e_{fj}	Comments
1	8.0 E 18	.5	71,000	¢
2	4.5 E 11	.5	35,000	Ref. 3.11
3	1.0 E 21	-1.5	224,900	3.12
4	3.6 E 18	-.82	103,000	3.13
5	2.8 E 12	.5	313,000	3.13
6	2.9 E 12	.5	333,000	3.13
7	2.2 E 20	-1.0	131,800	3.14
8	8.4 E 18	.5	120,000	¢
9	9.5 E 18	.5	140,000	¢
10	9.5 E 18	.5	117,000	¢
11	8.5 E 19	-1.0	257,900	3.14
12	1.0 E 19	.5	190,000	¢
13	9.3 E 18	.5	155,000	¢
14	9.4 E 18	.5	265,000	¢
15	9.5 E 18	.5	165,000	¢
16	9.5 E 18	.5	145,000	¢

¢ Activation energy computed using the method described by Semenov (Ref. 3.15), and the frequency factor computed using collision theory (Ref. 3.16).

were the best experimental values or were computed theoretically. With these constants, $\mathbb{R}$ = 1.987 cal/gmole - °K and temperature in °K, the units of f_j are cm^3/gm - mole - sec.

Originally the method proposed to obtain the equilibrium constants for the reactions in Table 3.5 was by a curve fit of equilibrium data. The chemical equilibrium composition for different values of temperature (from 1,000°k, to 15,000°K) were computed with a free energy minimization chemical equilibrium program. This information was then used with Eq. 3.37 to obtain the value of the equilibrium constant for different temperatures. The results of this operation are shown in Fig. 3.1 for the reaction $C_3H \rightleftarrows C_2H + C$ where ln K is plotted versus 1/T (dotted line). These actual values of K were then least squares fit with an expression of the form of Eq. 3.39. The resulting analytical expression for K is also plotted in Fig. 3.1 (solid line). It can be seen that using the analytical expression for K instead of the actual values introduces significant error, for example at T = 10,000°K the percentage error in K is 1000. This difficulty encountered in this reaction was found to be typical in some of the other reactions as well. For this reason it was found more desirable to use the second method discussed above for obtaining K_j. This was accomplished by obtaining $\Delta F_j{}^\circ$ from Eq. 3.41 and then using Eq. 3.40 to obtain K_j. Table 3.9 contains the values of the coefficients $B_{1,j}$, $B_{2,j} \ldots, B_{7,j}$ of Eq. 3.41 for the reactions in Table 3.5. The equilibrium constant K_j obtained, if plotted in Fig. 3.1, is undistinguishable from the curve of the actual values of K.

Radiative Properties

The radiation flux divergence $(dq_{R,y}/dy)$ appearing in Eq. 2.61

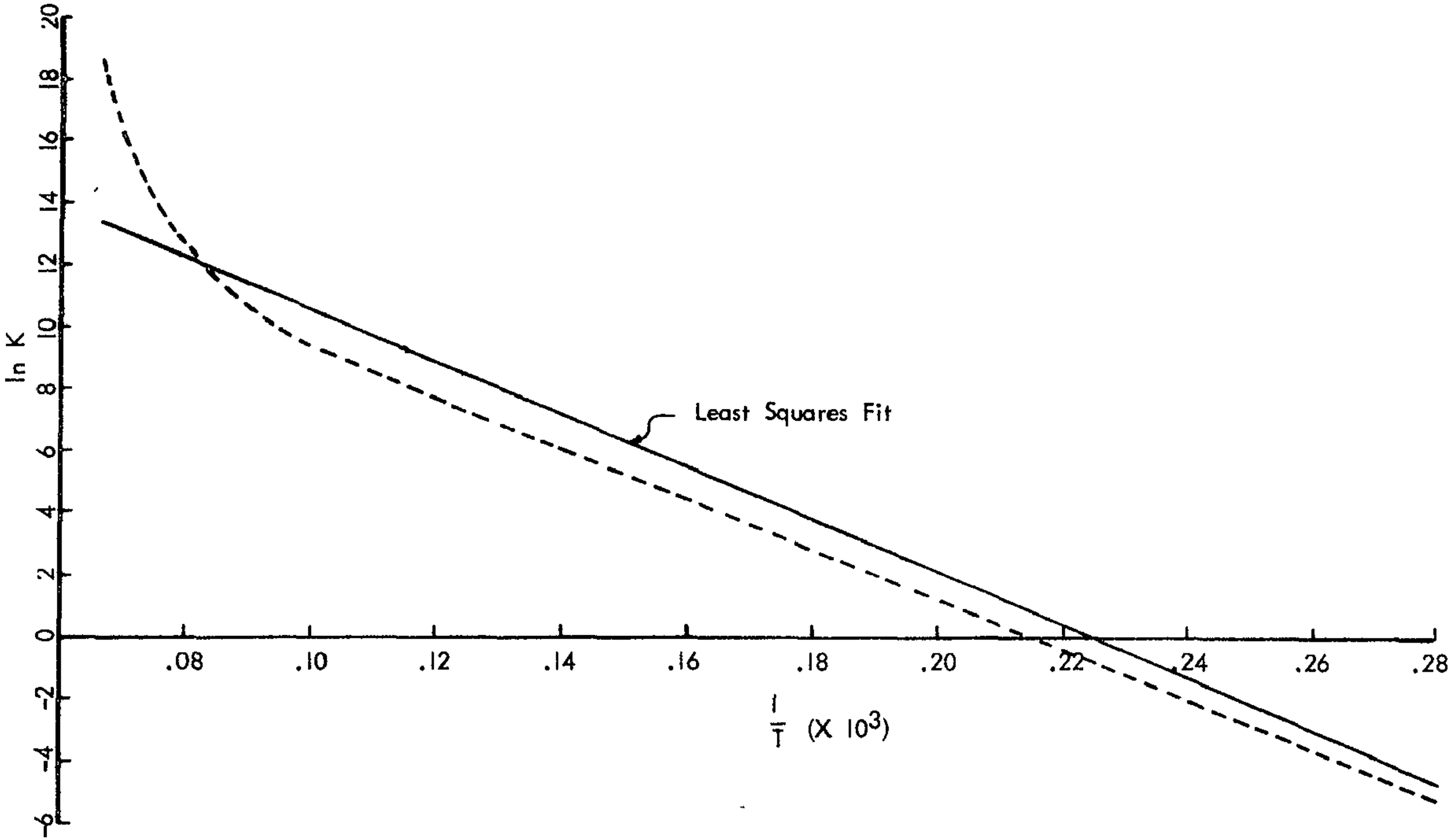


Figure 3.1 Equilibrium Constant and Its Linear Least Squares Fit For The Reaction $C_3H \rightleftharpoons C_2H + C$

TABLE 3.9

COEFFICIENTS OF THE EQUILIBRIUM CONSTANT CORRELATION

REACTION(*)	$B_{1,j}$	$B_{2,j}$	$B_{3,j}$	$B_{4,j}$	$B_{5,j}$	$B_{6,j}$	$B_{7,j}$	T(**)
1	0.3530E+00	-0.7681E-03	0.5153E-06	-0.9618E-10	0.5427E-14	0.3485E+05	0.7500E-01	L
	-0.9100E-01	0.2624E-03	-0.8700E-09	-0.5441E-11	0.2100E-15	0.4160E+05	0.2066E+01	H
2	0.1816E+01	-0.3571E-02	0.1634E-05	-0.2783E-09	0.1617E-13	0.1329E+05	-0.8961E+01	L
	-0.1852E+01	0.1831E-02	-0.5850E-06	0.7641E-10	-0.3417E-14	0.1329E+05	0.1126E+02	H
3	0.1727E+01	-0.8059E-03	0.1344E-06	0.4980E-11	-0.9780E-15	0.1132E+06	0.4274E+01	L
	0.1765E+01	-0.1250E-02	0.3816E-06	-0.3536E-10	0.1003E-14	0.1132E+06	0.4450E+01	H
4	0.1642E+01	-0.2810E-03	-0.9244E-07	0.2914E-10	-0.2112E-14	0.5196E+05	0.2626E+01	L
	0.4505E+01	-0.4018E-02	0.1254E-05	-0.1592E-09	0.7013E-14	0.5196E+05	-0.1348E+02	H
5	0.2321E+01	0.2250E-03	-0.9094E-07	0.1275E-10	-0.5056E-15	0.1580E+06	-0.1181E+02	L
	0.2904E+01	-0.3576E-03	0.6591E-07	-0.3159E-11	0.2813E-16	0.1580E+06	-0.1538E+02	H
6	0.2753E+01	-0.3726E-03	0.1884E-06	-0.3765E-10	0.2271E-14	0.1686E+06	-0.1238E+02	L
	0.2261E+01	0.3808E-03	-0.1210E-06	0.1070E-10	-0.3032E-15	0.1686E+06	-0.1000E+02	H
7	0.1675E+01	-0.6017E-03	-0.6914E-07	0.3996E-10	-0.2991E-14	0.9406E+05	0.3698E+01	L
	0.1414E+01	-0.8027E-03	0.1697E-06	-0.1315E-10	0.3831E-15	0.8731E+05	0.5594E+01	H
8	0.2257E+01	-0.2955E-02	0.1359E-05	-0.2518E-09	0.1668E-13	0.5850E+05	0.1912E+01	L
	0.3753E+01	-0.4486E-02	0.1768E-05	-0.2902E-09	0.1768E-13	0.6525E+05	-0.6819E+01	H

* SEE TABLE 3.5 FOR REACTICNS

** TEMPERATURE RANGE L= 1,000-6,000 DEG. K, H= 6,000-15,000 DEG. K

TABLE 3.9

COEFFICIENTS OF THE EQUILIBRIUM CONSTANT CORRELATION (Continued)

REACTION(*)	$B_{1,j}$	$B_{2,j}$	$B_{3,j}$	$B_{4,j}$	$B_{5,j}$	$B_{6,j}$	$B_{7,j}$	T(**)
9	0.3458E+01	-0.3852E-02	0.1541E-05	-0.2492E-09	0.1406E-13	0.6525E+05	-0.6335E+01	L
	0.2653E+01	-0.2187E-02	0.6688E-06	-0.8277E-10	0.3596E-14	0.6525E+05	-0.2220E+01	H
10	0.2094E+01	-0.2155E-02	0.7206E-06	-0.1067E-09	0.5735E-14	0.5766E+05	0.3671E+01	L
	0.2452E+01	-0.2382E-02	0.6930E-06	-0.8428E-10	0.3630E-14	0.5766E+05	0.1504E+01	H
11	0.2028E+01	-0.1370E-02	0.4461E-06	-0.5622E-10	0.2436E-14	0.1289E+06	0.3773E+01	L
	0.1323E+01	-0.5403E-03	0.1688E-06	-0.1859E-10	0.5931E-15	0.1289E+06	0.7660E+01	H
12	0.3053E+01	-0.4032E-02	0.1731E-05	-0.2858E-09	0.1621E-13	0.8906E+05	0.1034E+01	L
	-0.1596E+02	0.1840E-01	-0.5690E-05	0.6841E-09	-0.2842E-13	0.8906E+05	0.1101E+03	H
13	0.7810E+00	-0.1175E-03	-0.8460E-07	0.2854E-10	-0.2197E-14	0.7297E+05	0.9378E+01	L
	0.2560E+00	0.1581E-03	-0.3970E-07	0.2542E-11	0.2920E-17	0.7297E+05	0.1266E+02	H
14	0.2497E+01	0.6404E-04	-0.5011E-07	0.6619E-11	-0.2271E-15	0.1306E+06	-0.1216E+02	L
	0.2895E+01	-0.3234E-03	0.4932E-07	-0.2579E-11	0.2381E-16	0.1306E+06	-0.1481E+02	H
15	0.2132E+01	-0.2840E-02	0.1138E-05	-0.2021E-09	0.1335E-13	0.8068E+05	0.5461E+01	L
	0.3483E+01	-0.4981E-02	0.2072E-05	-0.3588E-09	0.2234E-13	0.8068E+05	-0.1881E+01	H
16	0.7030E+00	-0.1406E-02	0.5735E-06	-0.8895E-10	0.4609E-14	0.7220E+05	0.1162E+02	L
	0.2320E+00	-0.8811E-03	0.4090E-06	-0.6840E-10	0.3694E-14	0.7220E+05	0.1435E+02	H

* SEE TABLE 3.5 FOR REACTIONS

** TEMPERATURE RANGE L= 1.000-6.000 DEG. K. H= 6.000-15.000 DEG. K

is defined as follows (see Eq. 2.14)

$$\frac{dq_{R,y}}{dy} = -2\Pi \int \alpha_v \, (2B_v - I_v) \, dv \qquad (3.44)$$

where

$q_{R,y}$ = radiative flux in a direction normal to the body

α_v = volumetric absorption coefficient,

B_v = Plankian radiation intensity,

I_v = spectral radiation intensity,

v = frequency

Radiative mechanisms in high temperature gases may be categorized into those which produce radiation of a given frequency and those which produce radiation over a wide spectrum. The first of these groups of mechanisms is known as a line radiation mechanism (results from electronic transition between the bound energy levels in atoms or molecules), while the second group are commonly referred to as continuum radiation mechanisms (transfer between ionic states for atomic and molecular species, and transitions between two free energy levels in which free electrons are present in both the initial and final states).

One consequence of the existence of line and continuum radiative mechanisms is that the volumetric absorption coefficient (α_v) for a high temperature gas varies discontinuously with wavelength. This fact requires that the integration of Eq. 3.44 be carried out on a piecewise basis over the frequency domain. The frequency range is divided into regions (bands) within which the discontinuous vairations are averaged. Continuum radiation bands are used to represent regions of continuous radiation while line radiation bands are used to model the effect of the various discontinuous contributions. As in the numerical

integration of continuous functions, the use of more bands (smaller intervals) leads to a more accurate representation of the radiative process.

The radiative model used in the present work was developed by Engel (Ref. 3.17) from a model originally developed by Wilson (Ref. 3.18). The computer program developed by Engel computes both the radiative flux ($q_{R,y}$), and the radiative flux divergence ($dq_{R,y}/dy$) for a mixture of air and ablation products. The model uses nine line frequency bands and twelve continuum bands. It considers the following species and radiation mechanisms:

H		O_2 CO	
C	Line and	N_2 C_3	Continuum
O	Continuum	C_2 C_2H	
N		H_2	

REFERENCES

3.1 Esch, D.D., A. Siripong, and R.W. Pike, "Thermodynamic Properties in Polynomial Form for Carbon, Hydrogen, Nitrogen, and Oxygen Systems from 300 to 15,000°K," NASA-RFL-TR-70-3, Reacting Fluids Laboratory, Department of Chemical Engineering, Louisiana State University, Baton Rouge, Louisiana (November 1970).

3.2 Howerton, M.D., Engineering Thermodynamics, D. Van Nostrand Company Inc., Princeton, New Jersey, 1962, p. 138.

3.3 Chapman, S., and T.G. Cowling, The Mathematical Theory of Non-Uniform Gases, 2nd Edition, Cambridge University Press (1970).

3.4 Esch, D.D., Stagnation Region Heating of a Phenolic-Nylon Ablator During Return from Planetary Missions, Ph.D. Dissertation, Louisiana State University, Baton Rouge, Louisiana (1971).

3.5 Buddenberg, J.W., and C.R. Wilke, Ind. Eng. Chem., Vol. 41, pp. 1345-1347 (1949).

3.6 Williams, F.A., <u>Combustion Theory</u>, Addison - Wesley Publ shing Company, Inc., (1965).

3.7 Levenspiel, O., <u>Chemical Reaction Engineering</u>, John Wiley and Sons, Inc., New York (1962),

3.8 <u>Ibid</u>., p. 213.

3.9 <u>Ibid</u>., p. 213.

3.10 Esch, D.D., C.D. Engel, R.C. Farmer, R.W. Pike, G. Perez, and J.F. Balhoff, "Preliminary Assessment of the Effect of Finite-Rate Chemistry and Diffusion on Radiant Heating in Ablating Protection Systems," NASA-RFL-TR-70-4, Reacting Fluids Laboratory, Department of Chemical Engineering, Louisiana State University, Baton Rouge, Louisiana (December, 1970).

3.11 Eschenroeder, A.Q., and J.A. Lordi, "Catalysis of Recombination in Non-Equilibrium Nozzle Flows," <u>Ninth Symposium on Combustion</u>, Academic Press, New York (1963), p. 241.

3.12 Bahn, G.S., <u>Reaction Rate Compilation for the H-O-N System</u>, Gordon and Breach Science Publishers, New York, (1968), p. 23.

3.13 Pike, R.W.,"Evaluation of the Literature for Chemical Reactions and the Reaction Rates for the Decomposition Products from Charring Ablators," Langley Working Paper, LWP-181, NASA (January 21, 1960).

3.14 Menees, G.P., and R.L. McKenzie, "A Simplified Chemical Model for Estimating Non-Equilibrium Radiant Emission of CN (Violet) in Shock Heated Mixtures of CO_2 and N_2," <u>AIAA Journal</u>, Vol. 6, No. 3, p. 554 (1968).

3.15 Semenov, N.N., <u>Some Problems of Chemical Kinetics and Reactivity</u>, Vol. I p.2, Pergamon Press, New York (1958).

3.16 Levenspiel, <u>op. cit.</u>, p. 24.

3.17 Engel, Carl D., Ablation and Radiation Coupled Viscous Hypersonic Shock Layers, Ph.D. Dissertation, Louisiana State University, Baton Rouge, Louisiana (1971).

3.18 Wilson, K.H., "Stagnation Point Analysis of Coupled Viscous-Radiating Flow with Massive Blowing," Lockheed Missiles and Space Company, Palo Alto Research Laboratory, LMSC-687209 (April, 1969).

CHAPTER 4

THE SOLUTION METHOD

The mathematical model of stagnation-line flow contains a series of assumptions designed to simplify the mathematics of the problem without significantly reducing the accuracy of the solution. However, the resulting equations are still non-linear, integral, and coupled and have variable coefficients, therefore an analytical solution cannot be obtained. This means that integration of the equations must be accomplished numerically. Before an attempt is made to formulate a numerical solution scheme, it is convenient to nondimensionalize and transform the equations of the stagnation line model (Table 2.9). After the model has been transformed, the behavior of specific terms, equations, and the entire system of equations will be investigated so that a solution method may be established. The solution method will then be implemented with a computer program.

NONDIMENSIONALIZATION AND TRANSFORMATION

Using the dimensionless variables given in Table 4.1, the stagnation line model can be written in the dimensionless form given in Table 4.2. Notice that the only difference between the dimensionless variables given in Table 4.1 and those used for the order of magnitude assesment in Chapter 2 (Table 2.2) is that in the former $\rho_{\delta,o}{}^*$ is used in place of $\rho_\infty{}^*$. This has the effect of introducing into the momentum equation (Eq. 4.2) a Reynolds Number based upon post-shock density as opposed to the free-stream

TABLE 4.1

DIMENSIONLESS VARIABLES

$u = \dfrac{u^*}{U^*_\infty}$	$v = \dfrac{v^*}{U^*_\infty}$	$P = \dfrac{P^*}{\rho^*_{\delta,o}\ U^{*2}_\infty}$
$\rho = \dfrac{\rho^*}{\rho^*_{\delta,o}}$	$x = \dfrac{x^*}{R^*}$	$y = \dfrac{y^*}{R^*}$
$T = \dfrac{T^*}{T^*_{\delta,o}}$	$h = \dfrac{h^*}{\frac{1}{2}\ U^{*2}_\infty}$	$h_i = \dfrac{h^*_i}{\frac{1}{2}\ U^{*2}_\infty}$
$Cp = \dfrac{T^*_{\delta,o}\ C^*_p}{\frac{1}{2}\ U^{*\,2}_\infty}$	$D = \dfrac{D^*}{R^*\ U_\infty^*}$	$\mu = \dfrac{\mu^*}{\mu^*_{\delta,o}}$
$k = \dfrac{T_{\delta,o}^*\ k^*}{R^*\ P_{\delta,o}^*\ U^{*3}}$	$\omega_i = \dfrac{R^*\ \omega_i^*}{P_{\delta,o} U^*}$	$J_i = \dfrac{J_i^*}{\rho^*_{\delta,o}\ U^*_\infty}$
$q_{R,y} = \dfrac{q_{R,y}^*}{\rho^*_{\delta,o}\ U^{*3}_\infty}$	$\dfrac{\mathbf{R}}{M_i} = \dfrac{T_{\delta,o}^*}{U^{*2}_\infty}\ \dfrac{\mathbf{R}^*}{M_i^*}$	$R_{e_\delta} = \dfrac{\rho^*_{\delta,o}\ U^*_\infty\ R^*}{\mu^*_{\delta,o}}$
$\bar{\rho} = \dfrac{\rho_\infty^*}{\rho^*_{\delta,o}}$	$\delta = \dfrac{\delta^*}{R^*}$	

TABLE 4.2

DIMENSIONLESS STAGNATION LINE MODEL

EQUATIONS:

Species Continuity:

$$\frac{d}{dy}\left(\rho D \frac{dC_i}{dy}\right) - \rho v \frac{dC_i}{dy} + \omega_i = 0 \qquad (i = 1,\ldots,n) \tag{4.1}$$

X-momentum:

$$\frac{1}{R_{es}} \frac{d}{dy}\left[\mu \frac{d\left(\frac{1}{\rho}\frac{d(\rho v)}{dy}\right)}{dy}\right] - \rho v \frac{d}{dy}\left(\frac{1}{\rho}\frac{d(\rho v)}{dy}\right)$$

$$+ \frac{1}{2}\rho\left(\frac{1}{\rho}\frac{d(\rho v)}{dy}\right)^2 - 4\bar{\rho}(1-\bar{\rho})\left(\frac{d\phi}{dx}\right)^2_{x=0} = 0 \tag{4.2}$$

Energy:

$$\frac{2k}{C_p}\frac{d^2h}{dy^2} + \left[-\rho v + \frac{d\left(\frac{2k}{C_p}\right)}{dy}\right]\frac{dh}{dy} = \frac{2dq_{R,y}}{dy}$$

$$+ 2\rho v^2 \frac{dv}{dy} + \frac{d\left[\left(\frac{2k}{C_p} - \rho D\right)\left(\sum_i^n h_i \frac{dC_i}{dy}\right)\right]}{dy} \tag{4.3}$$

Caloric Equation of state:

$$h = \sum_i^n C_i h_i\ (T) \tag{4.4}$$

TABLE 4.2 (CONTINUED)

Thermal Equation of State:

$$\rho \, \mathbb{R} \, T \left(\sum_{i}^{n} \frac{C_i}{M_i}\right) = P_\delta \tag{4.5}$$

Boundary Conditions:

Wall $(y = 0)$

1. $\rho v = (\rho v)_w$
2. $\frac{d(\rho v)}{dy} = 0$
3. $C_i = C_{i,w} \; (i = 1, \ldots, n)$
4. $h = h_w$

Shock $(y = \delta)$

1. $\rho v = (\rho v)_\delta = v_\delta$
2. $\frac{d(\rho v)}{dy} = -2 \left(\frac{\partial u_\delta}{\partial x}\right)_{x=0}$
3. $C_i = C_{i,\delta} \; (i = 1, \ldots, n)$
4. $h = h_\delta$

(4.6)

Reynolds number previously used.

The mathematical model, as originally developed in Chapter 2 (Table 2.9), included the energy equation in temperature form (Eq. 2.61). During the present investigation the early attempts at solving the model concentrated on this formulation. However, after much computational experimentation led consistently to failure, it was decided to try a different formulation of the energy equation. The energy equation in terms of dimensionless temperature

$$\frac{d(k\,\frac{dT}{dy})}{dy} - \frac{1}{2}\,\rho v C_p\,\frac{dT}{dy} + \frac{1}{2}\,\frac{d[\rho D(\sum_i^n hi\,\frac{dC_i}{dy})]}{dy}$$

$$- \rho v^2\,\frac{dv}{dy} - \frac{1}{2}\,\rho v\,(\sum_i^n hi\,\frac{dC_i}{dy}) - \frac{dq_{R,y}}{dy} = 0 \tag{4.3a}$$

was transformed to an enthalpy form by nondimensionalizing Eq. 2.60 to yield

$$\frac{dT}{dy} = \frac{1}{C_p}\,\frac{dh}{dy} - \frac{1}{C_p}\,\sum_i^n hi\,\frac{dC_i}{dy} \tag{4.3b}$$

and substituting Eq. 4.3b in Eq. 4.3a to yield Eq. 4.3 (see Table 4.2). The model with the temperature form of the energy equation consisted of 3 + n equations in the unknowns ρ, v, T and C_i ($i = 1,\ldots,n$). The modified model includes the enthalpy as an additional unknown (temperature continues to be an unknown since the thermodynamic, transport, radiative and chemical kinetics properties are explicit function of T), and therefore requires an additional equation. This equation is the caloric equation of state (Eq. 4.4). The boundary conditions on the energy equation are now $h = hw$ at $y = 0$ and $h = h_\delta$ at $y = \delta$.

The equations given in Table 4.2 can now be transformed using the Dorodnitsyn transformation defined as:

$$\eta = \frac{\int_o^y \rho\, dy}{\int_o^\delta \rho\, dy} = \frac{\int_o^y \rho\, dy}{\tilde{\delta}} \tag{4.7}$$

From the above equation

$$\frac{d}{dy} = \frac{\rho}{\tilde{\delta}} \frac{d}{d\eta} \tag{4.8}$$

and this relation yields the equations given in Table 4.3. Notice that the Dorodnitsyn transformation converts the independent variable y to the new independent variable η. Before proceeding to discuss the numerical solution to Eqs. 4.9 - 4.11 and Eqs. 4.4 and 4.5 it is convenient to rewrite the equations in a form more suitable for numerical solution.

Species continuity:

Expanding the second order term in Eq. 4.9 and rearranging gives

$$\rho^2 D \frac{d^2 C_i}{d\eta^2} + \left[\frac{d(\rho^2 D)}{d\eta} - \tilde{\delta}\, \rho v \right] \frac{dC_i}{d\eta} + \frac{\tilde{\delta}^2}{\rho}\, \omega_i = 0 \quad (i = 1,\ldots,n) \tag{4.13}$$

X-momentum:

Let us define a new dependent variable as follows

$$f = \frac{\rho v}{\left(\frac{d(\rho v)}{d\eta}\right)_{\eta = 1}} \tag{4.14}$$

therefore

$$\frac{df}{d\eta} = \frac{\frac{d(\rho v)}{d\eta}}{\left(\frac{d(\rho v)}{d\eta}\right)_{\eta = 1}} \tag{4.15}$$

When the global continuity equation is nondimensionalized and transformed into η - space the result is

$$2\left(\frac{\partial u}{\partial x}\right)_{x = 0} = -\frac{1}{\tilde{\delta}} \frac{d(\rho v)}{d\eta} \tag{4.16}$$

TABLE 4.3

TRANSFORMED STAGNATION LINE MODEL

EQUATIONS:

Species Continuity:

$$\frac{d\left(\rho^2 D \frac{dC_i}{d\eta}\right)}{d\eta} - \tilde{\delta}\, \rho v \frac{dC_i}{d\eta} + \frac{\tilde{\delta}^2}{\rho}\, \omega_i = 0 \quad (i = 1,\ldots,n) \tag{4.9}$$

X-momentum:

$$\frac{d\left[\rho\mu \frac{d^2(\rho v)}{d\eta^2}\right]}{d\eta} - \tilde{\delta}\, R_{e_\delta}\, \rho v\, \frac{d^2(\rho v)}{d\eta^2} \tag{4.10}$$

$$+ \frac{\tilde{\delta}}{2} R_{e_\delta} \left(\frac{d(\rho v)}{d\eta}\right)^2 - 4\tilde{\delta}^3 R_{e_\delta} \frac{\bar{\rho}}{\rho} (1-\bar{\rho}) \left(\frac{d\phi}{dx}\right)^2_{x=0} = 0$$

Energy:

$$\frac{2}{\tilde{\delta}} \frac{\rho k}{C_p} \frac{d^2 h}{d\eta^2} + \left[-\rho v + \frac{1}{\tilde{\delta}} \frac{d\left(\frac{2\rho k}{C_p}\right)}{d\eta}\right] \frac{dh}{d\eta} =$$

$$2\frac{\tilde{\delta}}{\rho} \frac{dq_{R,y}}{dy} + 2\rho v^2 \frac{dv}{d\eta} + \frac{1}{\tilde{\tilde{\delta}}} \frac{d\left[\left(\frac{2\rho k}{C_p} - \rho^2 D\right)\left(\sum_i^n h_i \frac{dC_i}{d\eta}\right)\right]}{d\eta} \tag{4.11}$$

Caloric Equation of State:

$$h = \sum_i^n C_i H_i \tag{4.4}$$

TABLE 4.3 (CONTINUED)

Thermal Equation of State:

$$\rho \, \mathbb{R} \, T \left(\sum_{i}^{n} \frac{C_i}{M_i}\right) = P_\delta \tag{4.5}$$

Boundary Conditions:

Wall ($\eta = 0$)

1. $\rho v = (\rho v)_w$
2. $\frac{d(\rho v)}{d\eta} = 0$
3. $C_i = C_{i,w} \quad (i = 1,\ldots,n)$
4. $h = h_w$

Shock ($\eta = 1$)

1. $\rho v = (\rho v)_\delta = v_\delta$
2. $\frac{d(\rho v)}{d\eta} = -2\tilde{\delta} \left(\frac{\partial u_\delta}{\partial x}\right)_{x = 0}$
3. $C_i = C_{i,\delta} \quad (i = 1,\ldots,n)$
4. $h = h_\delta$

(4.12)

Substitution of Eq. 4.16 in Eqs. 4.14 and 4.15 gives

$$f = - \frac{\rho v}{2\tilde{\delta} \frac{\partial u_{\delta,o}}{\partial x}} \tag{4.17}$$

$$\frac{df}{d\eta} = \frac{\left(\frac{\partial u}{\partial x}\right)_{x=0}}{\left(\frac{\partial u_\delta}{\partial x}\right)_{x=0}} \tag{4.18}$$

Making use of L'Hospital's rule,

$$\lim_{x \to 0} \frac{u}{u_\delta} = \frac{\left(\frac{\partial u}{\partial x}\right)_{x=o}}{\left(\frac{\partial u_\delta}{\partial x}\right)_{x=o}} \tag{4.19}$$

and

$$\frac{df}{d\eta} = \lim_{x \to o} \frac{u}{u_\delta} \tag{4.20}$$

follows from Eqs. 4.18 and 4.19.

Substitution of Eq. 4.17 in Eq. 4.10 gives

$$\frac{d\left(\rho\mu \frac{d^2 f}{d\eta^2}\right)}{d\eta} + R_{e_\delta} \tilde{\delta}^2 \left(\frac{\partial u_{\delta,o}}{\partial x}\right) \left[2f \frac{d^2 f}{d\eta^2} - \left(\frac{df}{d\eta}\right)^2\right. \tag{4.21}$$

$$+ 2 \frac{\bar{\rho}}{\rho} (1 - \bar{\rho}) \frac{\left(\frac{d\phi}{dx}\right)^2_{x=o}}{\left(\frac{\partial u_{\delta,o}}{\partial x}\right)^2} = 0$$

The boundary conditions for this equation are

Wall ($\eta = o$):

1. $f = - \dfrac{(\rho v)w}{2\tilde{\delta} \frac{\partial u_{\delta,o}}{\partial x}}$

2. $\dfrac{df}{d\eta} = o$

Shock ($\eta = 1$):

1. $f = - \dfrac{v_\delta}{2\tilde{\delta} \frac{\partial u_{\delta,o}}{\partial x}}$

2. $\dfrac{df}{d\eta} = 1$

The momentum equation, being third order, can be expanded into two coupled equations; one first order and the other second order by defining

$$\mathcal{Z} = \frac{1}{\tilde{\delta}} \frac{df}{d\eta} \tag{4.22}$$

and substituting into Eq. 4.21

$$\rho\mu \frac{d^2\mathcal{Z}}{d\eta^2} + [2R_{e_\delta} \tilde{\delta}^2 \frac{\partial u_{\delta,o}}{\partial x} f + \frac{d(\rho\mu)}{d\eta}] \frac{d\mathcal{Z}}{d\eta}$$
$$-R_{e_\delta} \tilde{\delta}^3 \frac{\partial u_{\delta,o}}{\partial x} \mathcal{Z}^2 + 2R_{e_\delta} \tilde{\delta} \frac{\bar{\rho}}{\rho} (1 - \bar{\rho}) \frac{\left(\frac{d\phi}{dx}\right)^2_{x=0}}{\frac{\partial u_{\delta,o}}{\partial x}} = 0 \tag{4.23}$$

The resulting boundary conditions for Eqs. 4.22 and 4.23 are

Wall ($\eta = 0$):

1. $f = f_w = - \frac{(\rho v)w}{2\tilde{\delta} \frac{\partial u_{\delta,o}}{\partial x}}$

2. $\mathcal{Z} = 0$

Shock ($\eta = 1$):

1. $f = f_\delta = - \frac{v_\delta}{2\tilde{\delta} \frac{\partial u_{\delta,o}}{\partial x}}$

2. $\mathcal{Z} = 1/\tilde{\delta}$

As was discussed in Chapter 2, four boundary conditions need to be satisfied by the x - momentum equation, three because the equation is third order and the other since the shock stand-off distance (δ) is unknown. In terms of Eqs. 4.22 and 4.23, this means that these equations must satisfy all four of Eqs. 4.24. This is done in practice by applying the two boundary conditions on $\mathcal{Z}$ and f shown above to Eqs. 4.23 and 4.22, respectively. Integration of Eq. 4.22 gives

$$f = \tilde{\delta} \int_o^\eta \mathcal{Z}\, d\eta + f_w \tag{4.25}$$

Evaluating this relation at $\eta = 1$ and solving for $\tilde{\delta}$ yields

$$\tilde{\delta} = \frac{f_\delta - f_w}{\int_o^1 \mathcal{Z}\, d\eta} \tag{4.26}$$

It is easily seen that substitution of Eqs. 4.25 and 4.26 in Eq. 4.23 yields a second order, non-linear, integro-differential equation with Z as the only dependent variable (assuming that $\rho(\eta)$ and $\mu(\eta)$ are known). This equation can be solved for Z and then Eq. 4.25 is used to complete the solution to the problem. An alternative method of solution used in the present work is iteration: guess a value of $\tilde{\delta}$ and solve Eqs. 4.23 and 4.25 for f and Z, use the computed Z in Eq. 4.26 to obtain a new value of $\tilde{\delta}$ and repeat the above procedure using the new value of $\tilde{\delta}$ as guess until the guessed and computed vales of $\tilde{\delta}$ are equal.

Energy:

Expansion of the second term on the right-hand side of Eq. 4.11 gives

$$\frac{1}{\tilde{\delta}}\frac{d[(\frac{2\rho k}{C_p}-\rho^2 D)(\sum_i^n h_i\frac{dC_i}{d\eta})]}{d\eta}=\frac{1}{\tilde{\delta}}\frac{d[(\frac{2\rho k}{C_p}-\rho^2 D)(\sum_i^n h_i\frac{dC_i}{d\eta})]}{d\eta}$$

$$+\frac{1}{\tilde{\delta}}(\frac{2\rho k}{C_p}-\rho^2 D)(\sum_i^n\frac{dh_i}{d\eta}\frac{dC_i}{d\eta})+\frac{1}{\tilde{\delta}}(\frac{2\rho k}{C_p}-\rho^2 D)(\sum_i^n h_i\frac{d^2C_i}{d\eta^2}) \quad (4.27)$$

or

$$\frac{2}{\tilde{\delta}}\frac{\rho k}{C_p}\frac{d^2h}{d\eta^2}+[-\rho v+\frac{1}{\tilde{\delta}}\frac{d(\frac{2\rho k}{C_p})}{d\eta}]\frac{dh}{d\eta}=$$

$$\frac{2\tilde{\delta}}{\rho}\frac{dq_{R,y}}{dy}+2\rho v^2\frac{dv}{d\eta}+\frac{1}{\tilde{\delta}}\frac{d[(\frac{2\rho k}{C_p}-\rho^2 D)(\sum_i^n h_i\frac{dC_i}{d\eta})]}{d\eta}$$

$$+\frac{1}{\tilde{\delta}}(\frac{2\rho k}{C_p}-\rho^2 D)(\sum_i^n\frac{dh_i}{d\eta}\frac{dC_i}{d\eta})+\frac{1}{\tilde{\delta}}(\frac{2\rho k}{C_p}-\rho^2 D)(\sum_i^n h_i\frac{d^2C_i}{d\eta^2}) \quad (4.28)$$

A summary of the dimensionless transformed equations is given in Table 4.4.

METHODS OF COMPUTING REACTING FLOWS

Historically the first attempts at solving the finite-rate chemistry

TABLE 4.4

SUMMARY OF DIMENSIONLESS TRANSFORMED EQUATIONS

EQUATIONS:

Species Continuity:

$$\rho^2 D \frac{d^2 C_i}{d\eta^2} + \left[\frac{d(\rho^2 D)}{d\eta} - \tilde{\delta}\, \rho v \right] \frac{dC_i}{d\eta} + \frac{\tilde{\delta}^2}{\rho}\, \omega_i = 0 \quad (i = 1, \ldots, n) \tag{4.13}$$

X-Momentum:

$$f = - \frac{\rho v}{2\tilde{\delta}\, \dfrac{\partial u_{\delta,o}}{\partial x}} \tag{4.17}$$

$$\rho \mu \frac{d^2 Z}{d\eta^2} + \left[2R_{e_\delta} \tilde{\delta}^2 \left(\frac{\partial u_{\delta,o}}{\partial x}\right) f + \frac{d(\rho\mu)}{d\eta} \right] \frac{dZ}{d\eta} - R_{e_\delta} \tilde{\delta}^3 \left(\frac{\partial u_{\delta,o}}{\partial x}\right) Z^2 + 2R_{e_\delta} \tilde{\delta} \frac{\bar{\rho}}{\rho} (1 - \bar{\rho}) \frac{\left(\dfrac{d\phi}{dx}\right)^2_{x=0}}{\dfrac{\partial u_{\delta,o}}{\partial x}} = 0 \tag{4.23}$$

$$f = \tilde{\delta} \int_o^\eta Z \, d\eta + f_w \tag{4.25}$$

$$\tilde{\delta} = \frac{f_\delta - f_w}{\int_o^1 Z \, d\eta} \tag{4.26}$$

TABLE 4.4 (CONTINUED)

Energy:

$$\frac{2}{\tilde{\delta}}\frac{\rho k}{C_p}\frac{d^2h}{d\eta^2} + [-\rho v + \frac{1}{\tilde{\delta}}\frac{d(\frac{2\rho k}{C\rho})}{d\eta}]\frac{dh}{d\eta} =$$

$$\frac{2\tilde{\delta}}{\bar{\rho}}\frac{dq_{R,y}}{dy} + 2\rho v^2\frac{dv}{d\eta} + \frac{1}{\tilde{\delta}}\frac{d(\frac{2\rho k}{C_\rho} - \rho^2 D)}{d\eta}(\sum_i^n h_i \frac{dC_i}{d\eta})$$

$$+ \frac{1}{\tilde{\delta}}(\frac{2\rho k}{C_P} - \rho^2 D)(\sum_i^n \frac{dh_i}{d\eta}\frac{dC_i}{d\eta}) + \frac{1}{\tilde{\delta}}(\frac{2\rho k}{C_\rho} - \rho^2 D)(\sum_i^n h_i \frac{d^2C_i}{d\eta^2}) \qquad (4.28)$$

Caloric Equation of State:

$$h = \sum_i^n C_i h_i \qquad (4.4)$$

Thermal Equation of State:

$$\rho\, IR\, T\, (\sum_i^n \frac{C_i}{M_i}) = P_\delta \qquad (4.5)$$

problem were concentrated upon one-dimensional flows with no diffusion. The equations describing this physical situation are of the following form:

$$\frac{dy_i}{dt} = f_i\,(y_1,y_2,\ldots,y_n)$$

$$(i = 1,2,\ldots,n) \qquad (4.29)$$

$$y_i\,(o) = Y_{i,o}$$

where the concentrations of species in an n-species system $(y_1,\ldots,y_n)$ are the dependent variables and time (t) is the independent variable. These equations are non-linear, coupled through the right-hand-side term $(f_1,\ldots, f_n)$, and are subject to initial boundary conditions of the first kind. In attempting to numerically solve the above set of equations it was found that the classical integration techniques (Euler, Runge-Kutta, etc.) could be used to compute the solution for cases where the reaction rates were relatively slow, but for fast reaction rates (those occurring when the flow is near chemical equilibrium) the time required to integrate the equations became exhorbitant. When the latter condition existed the equations were referred to as "stiff" equations by Curtiss and Hirschfelder (Ref. 4,1) because this type of problem arises also in overcontrolled servomechanical systems.

Numerical techniques for solving differential equations are generally judged on the basis of stability, accuracy, and speed of computation. Stability means that the numerical solution must follow the general form of the true solution. Accuracy measures how closely the numerical solution approximates the true solution. It should be clear that an algorithm may be stable but inaccurate. The computational speed depends upon the allowable stepsize and on the computational effort required to perform each step.

The solution of Eqs. 4.29 is normally carried out by locally linearizing the equations and solving at each step the resulting set of linear equations. The linearization is carried out by expanding each f_i in a truncated Taylor series about a local point t_m to yield

$$\frac{dy_i}{dt} = f_{i,m} + \sum_{j=1}^{n} \left(\frac{\partial fi}{\partial yj}\right)_m (y_j - y_{j,m}) \quad (i = 1,\ldots,n) \qquad (4.30)$$

or

$$\frac{dy_i}{dt} = \sum_{j=1}^{n} A_{ij} Y_j + B_i \quad (i = 1,\ldots,n) \qquad (4.31)$$

where

$$A_{ij} = \left(\frac{\partial fi}{\partial yj}\right)_m \quad \text{and } Bi = F_{i,m} - \sum_{j=1}^{n} \left(\frac{\partial fi}{\partial y_j}\right)_m y_{i,m}$$

Equation 4.31 may be written as

$$\frac{d\bar{y}}{dt} = \bar{\bar{A}}\,\bar{y} + \bar{B} \qquad (4.32)$$

using matrix notation where $\bar{y} = [y_1, y_2,\ldots y_n]^T$,

$\bar{B} = [B_1, B_2,\ldots B_n]^T$ and

$$\bar{\bar{A}} = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \cdots & \frac{\partial f_1}{\partial y_n} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \cdots & \frac{\partial f_2}{\partial y_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial y_1} & \frac{\partial f_n}{\partial y_2} & \cdots & \frac{\partial f_n}{\partial y_n} \end{bmatrix}$$

Techniques for solving Eq. 4.32 are generally classified as explicit, implicit and locally exact. Explicit techniques are those for which the values of each of the dependent variables at any step may be expressed explicitly in terms of the dependent variables at preceeding steps. All

predictor-corrector methods, as well as all variants of the Runge-Kutta method are explicit. Implicit techniques are those for which the values of each of the variables at any step depends on itself and on the other dependent variables at that step as well as on the dependent variables at preceeding steps. An implicit technique when applied to Eq. 4.32 always results in a set of linear algebraic simultaneous equations which must be solved for the vector $\bar{y}$ at each step. Locally exact techniques solve Eq. 4.32 exactly to yield

$$y_i = \sum_{j=1}^{n} C_{ij} y_{ij} \exp(\lambda_j h) + y_{i,p} \quad (i = 1,\ldots,n) \tag{4.33}$$

where the C_{ij} are integration constants, the λ_j are the eigenvalues of $\bar{\bar{A}}$, h is the stepsize and $y_{i,p}$ are particular solutions to Eq. 4.32. This solution assumes that all the eigenvalues of $\bar{\bar{A}}$ are distinct.

All explicit techniques have finite stability boundaries, that is, they are stable for some stepsizes and unstable for others. Under conditions of stiffness, explicit techniques must use a very small stepsize in order to remain stable. On the other hand, they can be very accurate and require a modest degree of computational effort to compute a step. Some implicit techniques are unconditionally stable, in other words, the allowable stepsize is not restricted by stability requirements. However, their accuracy is generally limited and, because a set of linear algebraic equations must be solved, they require a fair amount of computational effort to perform a step. Locally exact methods are unconditionally stable and very accurate but, because the eigenvalues of $\bar{\bar{A}}$ must be computed for each step, require a large degree of computational effort to compute a step.

The numerical solution of stiff equations has been studied extensively and a number of techniques were proposed to cope with this problem

(Refs. 4.1 - 4.15).

Curtiss and Hirschfelder (Ref. 4.1) proposed an implicit technique for solving a single stiff equation. Emanuel (Ref. 4.2) studied the application of predictor-corrector and Runge-Kutta techniques to the chemical equations in near equilibrium flows. The locally exact method was applied by Moretti (Ref. 4.3) to the problem of combustion of hydrogen in air at constant pressure. He made a comparison of the computing time required using the locally exact method and the fourth order Runge-Kutta method.

Treanor (Ref. 4.5) developed an explicit method for solving nonlinear stiff equations. It assumes that the basic equations can be approximated by a linearized form, but not the form given in Eqs. 4.30. Instead they are linearized as follows:

$$\frac{dy_i}{dt} = -(P_i)_m\, y_i + (A_i)_m + (B_i)_m\, h + (C_i)_m\, h^2 \qquad (i = 1,\ldots,n) \qquad (4.34)$$

where the P_i, A_i, B_i and C_i are constants in a given step but are allowed to vary from step to step.

Lomax and Bailey (Ref. 4.6) and Bailey (Ref. 4.12) studied the solution to the problem of air flow behind a normal shock by applying different numerical techniques. The techniques utilized were the explicit modified Euler method, the implicit modified Euler method and Treanor's method.

Magnus and Schechter (Ref. 4.15) developed a unified theory of numerical techniques based on rational approximations to the exponential terms in Eqs. 4.33. They showed that a number of explicit and implicit techniques are really rational approximations of different order. The

implicit methods studied, which include the method of collocation and the subdomain method, were shown to be unconditionally stable and the subdomain method was applied to the problems of hydrogen-air combustion and dissociating air.

From the work that has been carried out it has become evident that there is no "best" method for solving Eqs. 4.29, The optimum method to be used depends upon the degree of "stiffness" of the equations. If the flow is essentially frozen, classical explicit techniques such as fourth order Runge-Kutta are better than any implicit or locally exact method because they would be accurate and faster. If the flow is near chemical equilibrium, an unconditionally stable implicit technique is optimum because the stepsize would be restricted only by the truncation error. In general, the locally exact technique is less convenient than the explicit or implicit methods, but it yields more information about the system of equations.

This discussion has assumed that the problem represented by Eqs. 4.29 is "well behaved", this means that the solution is inherently stable, and our discussion of stability was centered around what is called induced instability, that is, instability induced by the numerical technique used. However, it can also happen that the problem is "ill posed" or inherently unstable (Ref. 4.16). This can happen, for example, when the general solution to Eqs. 4.29 contains exponentially growing terms which must be suppresed to fulfill the boundary conditions imposed on the problem. For example, the general solution to the equation

$$\frac{dy}{dt} = y - t \tag{4.35}$$

is given by

$$y = C\,e^{t} + t + 1 \tag{4.36}$$

where C is an integration constant. If the initial condition is given as y (o) = 1, which makes C = 0 so that the solution is a simple linear term growing only slowly in comparison with the exponential term, any numerical technique for solving Eq. 4.35 will introduce the exponential term into the computed solution resulting in inherent instability.

This type of condition is common in problems described by differential equations of the type

$$D_i(x) \frac{d^2 y_i}{d\,x^2} + E_i(x) \frac{dy_i}{dx} \qquad (i = 1,\ldots,n) \qquad (4.37)$$

$$= F_i(Y_1,\ Y_2,\ldots Y_n)$$

where the D_i and E_i are functions of the independent variable x; the F_i are nonlinear functions of $y_1,\ y_2,\ldots,y_n$, and Eqs. 4.37 are subject to the following boundary conditions

$$y_i(o) = y_{i,o} \qquad (i = 1,\ldots,n) \qquad (4.38)$$

$$Y_i(1) = y_{i,1} \qquad (i = 1,\ldots,n) \qquad (4.39)$$

One physical situation described by Eqs. 4.37 - 4.39 is that arising from similar boundary layer flows including mass diffusion and finite-rate chemical reactions.

When the equations are inherently unstable the methods discussed above for solving Eqs. 4.29 are not effective. Such methods are usually referred to as "marching" techniques because the solution starts at one of the boundaries and, for each step taken, the solution marches towards the other boundary. When marching techniques are applied to a two-point boundary value problem, such as that represented by Eqs. 4.37 - 4.39, it is necessary to transform the problem into an initial value problem.

This is accomplished by guessing the value of the $\frac{dy_i}{dx}$ at $x = 0$ and using, the marching technique to compute the y_i at $x = 1$, if the computed y_i at $x = 1$ differs from the desired values, it is necessary to iterate until the boundary conditions at $x = 1$ are met.

Because of the problems associated with the use of marching techniques, boundary layer flow problems including mass diffusion and finite-rate chemical reactions have been approached with what are known as "globally implicit" techniques. These are techniques which work directly with the two-point boundary value problem and produce in one step the solution over the domain of interest.

Fay and Kay (Ref. 4.16) used a globally implicit method to solve the equations describing a laminar, dissociating, nitrogen boundary layer at the stagnation point of an axisymmetric body. The boundary layer equations were first linearized about an initial trial solution (or previous iteration) in such a way that the equations are uncoupled (except implicitly through the trial solution). The equations were then cast into an implicit finite-difference form and solved in sequence, the trial solutions being updated after each solution had been obtained. This sequence of solutions was continued until the original differential equations were satisfied to the desired degree of accuracy.

Blottner (Ref. 4.17) studied techniques for solving the viscous shock layer flow at the stagnation point of a blunt body for air with finite-rate chemical reactions. He found that a globally implicit technique, such as the one used by Fay and Kay, produced converged solutions in a reasonable amount of time. Adams, et. al., (Ref. 4.18) applied the technique used by Fay and Kay to compute the inviscid and viscid flow fields around spherically blunted cone geometries, including injection of inert argon or chemically reacting carbon dioxide with chemical reactions

taking place at a finite -rate.

Liu (Ref. 4.19) studied the problem of hydrogen injection into air at an axisymmetric point including mass diffusion and non-equilibrium chemistry. He studied the applicability of various techniques, such as an implicit marching technique developed by Lomax (Ref. 4.10) and the globally implicit method used by Fay and Kay, and found that some of the methods could not be applied to the reacting flow problem and those that could be applied did not work for near chemical equilibrium flows.

To be successful, any proposed solution method to the stagnation-line, finite-rate equations must be in accord with these previous studies.

DECOUPLING THE EQUATIONS

Although it is desirable to solve the equations in Table 4.4 in a coupled manner, practical considerations make it necessary to decouple them. It must be noticed that any attempt to solve the equations coupled would require the simultaneous solution of 7 + n (in our problem n = 19) non-linear, ordinary integro-differential and linear integral and algebraic equations.

An illustration of how coupled equations can be uncoupled is given by considering the problem of solving

$$\frac{dL_1}{dt} = a_{11} L_1 + a_{12} L_2 \qquad (4.40)$$

$$\frac{dL_2}{dt} = a_{21} L_1 + a_{22} L_2 \qquad (4.41)$$

where L_1 and L_2 are the dependent variables and the a_{ij}'s are constants. If the function L_2 is guessed, $(L_2(o))$, then this value can be used in Eq. 4.40 to give

$$\frac{dL_1^{(1)}}{dt} - a_{11}L^{(1)} = a_{12}L_2^{(o)} \qquad (4.42)$$

where $L_2^{(o)}$ is a known function of t. This equation can be solved for $L_1^{(1)}$ and this function can be used as a guess for L_1 in Eq. 4.41 to give

$$\frac{dL_2^{(1)}}{dt} - a_{22} L_2^{(1)} = a_{21} L_1^{(1)} \tag{4.43}$$

The function $L_2^{(1)}$ obtained from solving the equation above can then be compared to $L_2^{(o)}$; if they are equal the problem is solved, if not $L_2^{(1)}$ is used as the new $L_2^{(o)}$ and the procedure described above is repeated. It is easy to see that the price that must be paid for reducing the original problem to that posed by Eqs. 4.42 and 4.43 is, in general, a greater amount of computation needed.

The scheme used to decouple the equations in Table 4.4 is given in Figure 4.1 while Table 4.5 contains the uncoupled equations. First of all, values of $\tilde{\delta},^{(o)}$, $\rho^{(o)}$, $(\rho\mu)^{(o)}$, and $C_i^{(o)}$ ($i = 1,\ldots,n$) are guessed. With these functions it is possible to solve Eqs. 4.44 and 4.45 for $f^{(1)}$ and $z^{(1)}$ (this section of the flowchart is that enclosed by the dotted lines), this is accomplished by guessing $f^{(o)}$ and solving Eq. 4.44 for $z^{(1)}$, then Eq. 4.45 is solved for $f^{(1)}$. If $f^{(1)} = f^{(o)}$ we proceed to solve for $v^{(1)}$ and $\delta^{(1)}$, if not, $f^{(o)}$ and $f^{(1)}$ are used to compute the next guess ($f^{(o)} = (1 - \lambda_1) f^o + \lambda_1 f^{(1)}$, where $0 < \lambda_1 \leq 1$) and the process is repeated. Once $f^{(1)}$ and $z^{(1)}$ are known, Eqs. 4.46 and 4.47 are solved for $v^{(1)}$ and $\tilde{\delta}^{(1)}$, respectively. If $\tilde{\delta}^{(1)} = \tilde{\delta}^{(o)}$ the iterative process

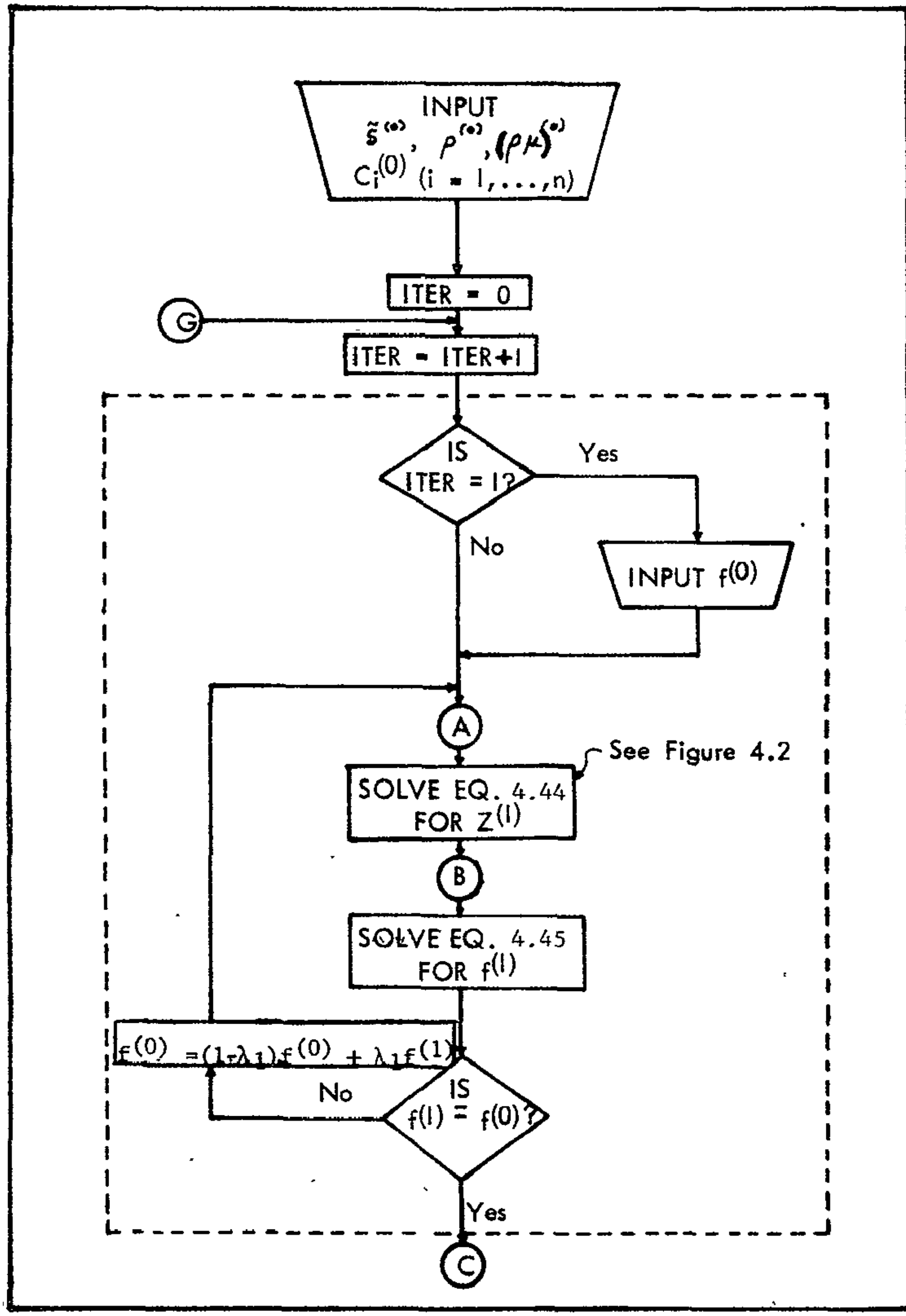


Figure 4.1 Problem Flowchart

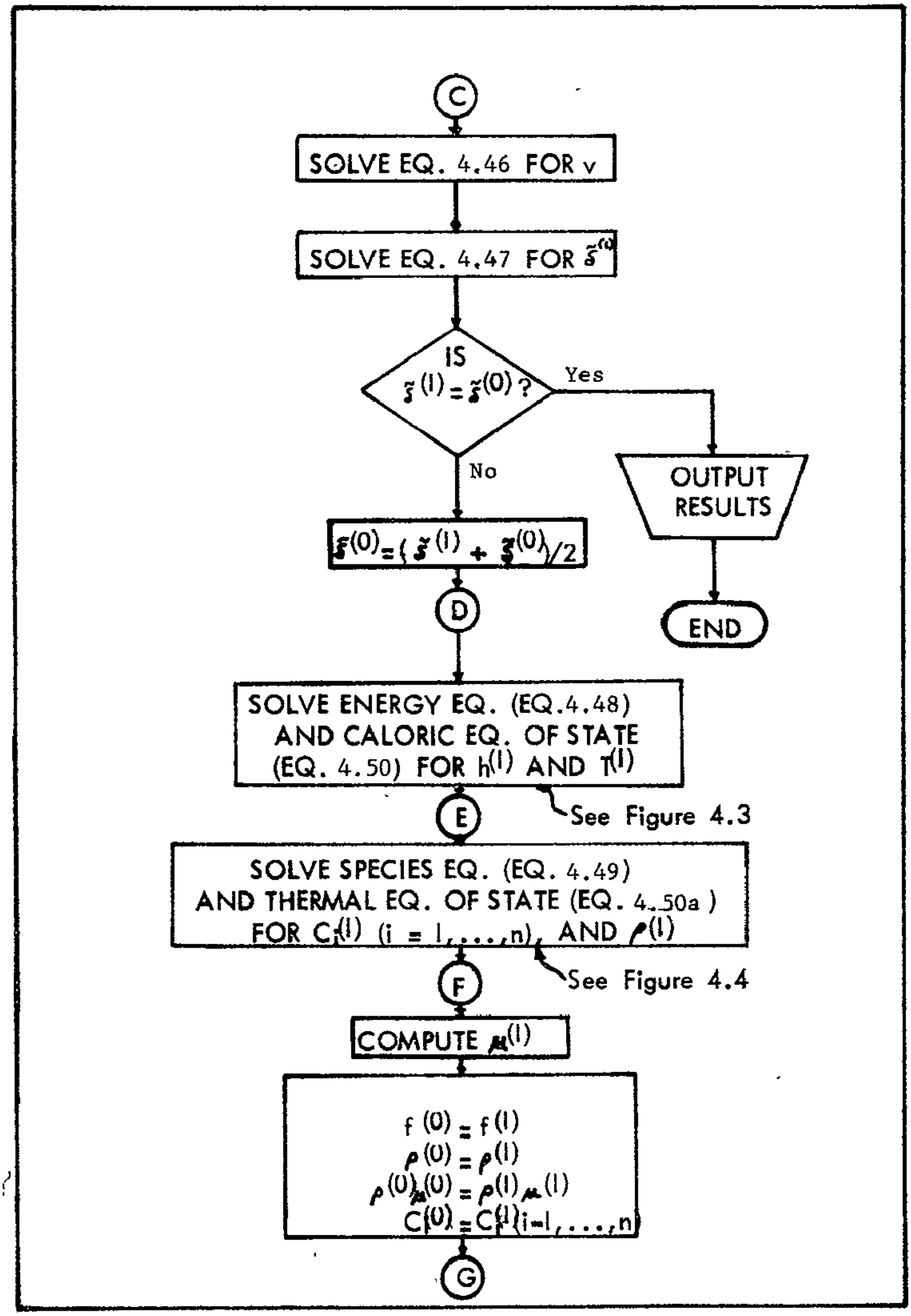


Figure 4.1 Problem Flowchart (continued)

TABLE 4.5

DECOUPLED EQUATIONS

χ - Momentum

$$\rho^{(0)}\mu^{(0)} \frac{d^2 z^{(1)}}{d\eta^2} + \left[2\, Re_\delta\, \tilde{\delta}^{(0)^2} \left(\frac{\partial u_{\delta,0}}{\partial \chi}\right) f^{(0)} + \frac{d(\rho^{(0)}\mu^{(0)})}{d\eta}\right] \frac{dz^{(1)}}{d\eta}$$

$$-Re_\delta \tilde{\delta}^{(0)\,3} \left(\frac{\partial \mu_{\delta,0}}{\partial \chi}\right) z^{(1)^2} + 2\, Re_\delta\, \tilde{\delta}^{(0)} \frac{\bar{\rho}}{\rho^{(0)}} (1 - \bar{\rho}) \frac{\left(\frac{d\phi}{d\chi}\right)^2 \chi = 0}{\frac{\partial \mu_{\delta,0}}{\partial \chi}} = 0 \qquad (4.44)$$

$$f^{(1)} = \tilde{\delta}^{(0)} \int_0^\eta z^{(1)}\, d\eta + f\omega \qquad (4.45)$$

$$v^{(1)} = -2 \frac{\tilde{\delta}^{(0)}}{\rho^{(0)}} \left(\frac{\partial \mu_{\delta,0}}{\partial \chi}\right) f^{(1)} \qquad (4.46)$$

$$\tilde{\delta}^{(1)} = \frac{f_\delta - f_\omega}{\int_0^1 z^{(1)} d\eta} \qquad (4.47)$$

Energy:

$$\frac{2}{\tilde{\delta}^{(1)}} \left(\frac{\rho k}{c_p}\right)^{(0)} \frac{d^2 h^{(1)}}{d\eta^2} + \left[-\rho^{(0)} v^{(1)} + \frac{1}{\tilde{\delta}^{(1)}} \frac{d}{d\eta} \left(\frac{2\rho k}{c_p}\right)^{(0)} \right] \frac{dh^{(1)}}{d\eta} =$$

$$\frac{2\tilde{\delta}^{(1)}}{\rho^{(0)}} \frac{dq^{(0)}{}_{R,y}}{dy} + 2\rho^{(0)} (\upsilon^{(1)^2}) \frac{dv^{(1)}}{d\eta} + \frac{1}{\tilde{\delta}^{(1)}} \frac{d\left[\frac{2\rho k}{c_p} - \rho^2 D\right)^{(0)}}{d\eta} \qquad (4.48)$$

$$\left. \left(\sum_i^h h_i \frac{d\, Ci}{d\eta} \right)^{(0)} \right]$$

TABLE 4.5 (Continued)

$$+ \frac{1}{\tilde{\delta}^{(1)}} \left(\frac{2\rho k}{c_p} - \rho^2 D\right)^{(0)} \left(\sum_i^n \frac{dhi}{dn}\frac{dCi}{dn}\right)^{(0)} + \frac{1}{\tilde{\delta}^{(1)}}\left[\left(\frac{2\rho k}{c_p} - \rho^2 D\right)^{(0)}\right.$$

$$\left.\left(\sum_i^n hi \frac{d^2 Ci}{dn^2}\right)^0\right]$$

Species Continuity:

$$\rho^{(0)^2} D^{(1)} \frac{d^2 C_i^{(1)}}{d\, n^2} + \left[\frac{d(\rho^{(0)^2} D^{(1)})}{dn} - \tilde{\delta}^{(1)}\rho^{(0)} v^{(1)}\right] \frac{d\, C_i^{(1)}}{dn}$$

$$+ \frac{\tilde{\delta}^{(1)^2}}{\rho^{(0)}} \quad \omega_i^{(1)} = 0 \qquad (4.49)$$

Caloric Equation of State:

$$T^{(1)} = fn\, \{h^{(1)}, C1^{(0)}, \ldots, Cn^{(0)}\} \qquad (4.50)$$

Thermal Equation of State:

$$\rho^{(1)} = \frac{P_\delta}{\left(\sum_i^n \frac{C_i^{(1)}}{M_i}\right) IRT^{(1)}} \qquad (4.50a)$$

is ended and the results output, if not the values of $\rho^{(o)}$, $C_i^{(o)}$ ($i = 1, \ldots, n$), $v^{(1)}$ and $\tilde{\delta}^{(1)}$ are used in the energy equation (Eq. 4.48) to solve for $h^{(1)}$, and the caloric equation of state (Eq. 4.50) is solved for $T^{(1)}$. Then the species equation (Eq. 4.49) and the thermal equation of state (Eq. 4.50a) are solved for $C_i^{(1)}$ ($i = 1, \ldots n$) and $\rho^{(1)}$. Once the temperature and concentration profiles are known, the new viscosity ($\mu^{(1)}$) is computed, and $\tilde{\delta}^{(1)}$, $\rho^{(1)}$, $(\rho\mu)^{(1)}$ and $C_i^{(1)}$ ($i=1, \ldots, n$) can be used as guesses for the next iteration. The flowcharts for the operations noted in Figure 4.1 are given in Figures 4.2 - 4.4 to be developed below.

LINEARIZATION OF THE EQUATIONS

In the previous section the equations were decoupled (Table 4.5); this has the effect of making each equation independent from the rest except through the iterative process. However, the equations are still nonlinear with variable coefficients. In considering the kinds of numerical techniques that may be used to solve these equations, the fact that the boundary conditions are given on two different points in the flow-field suggests that "globally implicit" finite-difference approximations would be preferable to so-called "shooting" or "marching" techniques since the latter requires specifications of all boundary conditions on one point in the flow-field. As used in this work globally implicit means that each differential equation is substituted by finite-difference approximations over the entire flow-field, and this results in a set of simultaneous algebraic equations that when solved yield the value of the dependent variable at different points in the flow field. If this finite-difference technique is applied directly to the nonlinear equations the problem is reduced to solving a set of non-linear

algebraic equations; however, due to the present lack of efficient techniques for solving sets of non-linear algebraic equations it is more convenient to linearize the differential equations before they are written in finite-difference form. When this is done the problem is reduced to obtaining the solution to a set of linear algebraic equations, a much easier problem.

Consider a non-linear ordinary differential equation of the form

$$\frac{dL\{t\}}{dt} = g\ \{L,t\} \tag{4.51}$$

where g {L,t} is the non-linear term. It is evident that there are various ways of linearizing this equation; the simplest one being to use an approximation to the solution $L^{(o)}$ to write

$$\frac{dL^{(1)}}{dt} = g\ \{L^{(o)},\ t\} \tag{4.25}$$

This equation is now linear since $L^{(o)}$ is a known function of t, it may be solved for $L^{(1)}$ and if $L^{(1)} = L^{(o)}$ the answer has been found, if not $L^{(1)}$ is used as the new guess and the process is repeated. Another commonly used linearizing technique is to expand the non-linear term in Eq. 4.51 about the function $L^{(o)}$ in a Taylor series truncated after the second term to yield

$$g\ \{L^{(1)},\ t\} = g\ \{L^{(o)},\ t\} + \frac{\partial g\{L^{(o)},\ t\}}{\partial t}\ (L^{(1)} - L^{(o)}) \tag{4.53}$$

and by substitution of this relation in Eq. 4.51

$$\frac{dL^{(1)}}{dt} - \frac{\partial g\{L^{(0)}, t\}}{\partial t} L^{(1)} = g\{L^{(1)}, t\} - \frac{\partial g\{L^{(0)}, t\}}{\partial t} L^{(0)} \tag{4.54}$$

This technique is known as quasilinerization (Refs. 4.20 and 4.21), and has been used extensively to solve problems in fluid and orbital mechanics. It can be demonstrated that if the sequence $\{L^{(K)}\}$ converges, that is if $\lim_{K \to \infty} \{L^{(K)}\} = L$, it does it monotonically (for example if $L^{(0)} < L$ for all t monotonicity of convergence means $L^{(0)} \le L^{(1)} \le \ldots \le L$, and quadratically

$$\left| L^{(K+1)} - L^{(K)} \right| = N \left| L^{(K)} - L^{(K-1)} \right| \tag{4.55}$$

where N is a constant. Although these two properties make the technique extremely powerful, in practice the computation of $\partial g/\partial t$ may be very costly. Therefore, in some instances the simpler technique given by Eq. 4.52 may be preferable.

In the present application use will be made of both the simple linearization and quasilinearization techniques.

X - momentum:

The non-linear term in Eq. 4.44 is the $Z^{(1)^2}$ term, and quasilinerization yields

$$Z^{(1)^2} = Z^{(0)^2} + 2Z^{(0)} (Z^{(1)} - Z^{(0)})$$

$$= 2Z^{(0)} Z^{(1)} - Z^{(0)^2} \tag{4.56}$$

when this relation is substituted in Eq. 4.44 the result is

$$\frac{d^2 Z^{(1)}}{d\eta^2} + 2\left[\frac{Re_{\delta} \, \tilde{\delta}^{(0)^2}}{\rho^{(0)} \mu^{(0)}} \frac{\partial \Pi_{\delta 0}}{\partial \chi} f(Q) + \frac{d}{d\eta}\left(\ln (\rho^{(0)} \mu^{(0)})\right)\right] \frac{dZ^{(1)}}{d\eta}$$

$$-2\ \frac{Re_\delta\ \delta^{(0)3}}{\rho^{(0)}\ \mu^{(0)}}\ \ \frac{\partial\ \mu_{\delta\ 0}}{\partial\ \chi}\ \ z^{(0)}\ \ z^{(1)}\ \ = \tag{4.57}$$

$$-2\ \frac{Re_\delta\ \tilde{\delta}^{(0)}}{\rho^{(0)}\mu^{(0)}}\left[\frac{\bar{\rho}\ (1-\bar{\rho})}{\rho^{(0)}}\ \frac{\left(\frac{d\phi}{dx}\right)^2\ x=0}{\frac{\partial\mu_{\delta,\ 0}}{\partial\ \chi}}+\frac{\delta^{(0)^2}}{2}\ \frac{\partial\xi_{\delta,o}}{\partial\ \chi}\ \ z^{(0)^2}\right]$$

This equation is now linear and may be solved as shown in Figure 4.2.

Energy:

The energy equation (Eq. 4.48) is linear in $h^{(1)}$, and therefore linearization is not needed. Inversion of the caloric equation of state is accomplished as follows:

From Eq. 4.4

$$h^{(1)}\ = \sum_i^n\ C_i^{(0)}\ \ h_i\ \{T\} \tag{4.4}$$

therefore the problem consists of finding a root of the equation

$$G\{T\} = h^{(1)} - \sum_i^n\ C_i^{(0)}\ h_i\ \{T\}$$

Since the $h_i\{T\}$ are nonlinear functions of T, the Newton-Raphson technique was used:

$$G\ \{T^{(K+1)}\}\ =\ G\ \{T^{(K)}\} + \frac{dG}{dT}^{(k)}\ \ (T^{(K+1)} - T^{(K)})$$

$$T^{(K+1)} - T^{(K)}\ =\ -\ \frac{G\{T^{(K)}\}}{\frac{dG}{dT}^{(K)}}$$

$$\Delta T^{(K)}\ = -\ \frac{h^{(1)} - \sum_i^n\ C_i^{(0)}\ h_i\ (T^{(K)})}{-\sum_i^n C_i^{(0)}\ C_{\rho i}\ (T^{(K)})}\ = \frac{h^{(1)} - h^{(K)}}{C_\rho\ ^{(K)}}$$

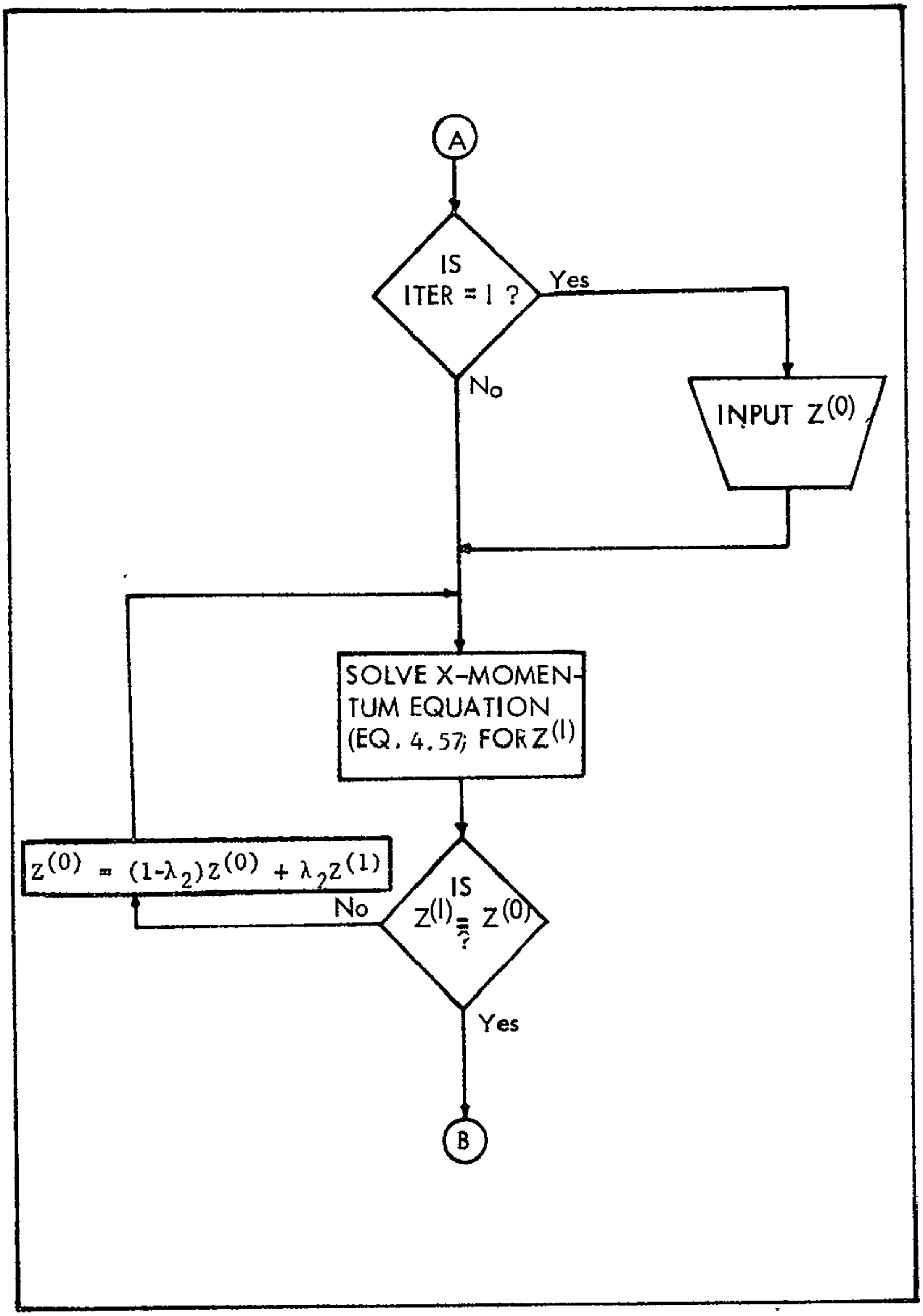


Figure 4.2 Solution of X-momentum Equation

Using underelexation to obtain convergence

$$T^{(K+1)} = T^{(K)} + \lambda_3 \, \Delta T^{(K)}$$

$$= T^{(K)} + \lambda_3 \; \frac{h^{(1)} - h^{(k)}}{C_\rho^{(K)}} \qquad 0 < \lambda_3 \leq 1 \qquad (4.58)$$

Once $T^{(K+1)}$ has converged it becomes the new $T^{(1)}$.

The flowchart for the solution of the equation of energy and the caloric equation of state for $h^{(1)}$ and $T^{(1)}$ is given in Fig. 4.3.

Species Continuity:

The reaction rate term in Eq. 4.49 is non-linear, therefore, quasilinearization gives

$$\omega_i^{(1)} = \omega_i^{(0)} + \frac{\partial \omega_i^{(0)}}{\partial C_i} \left(C_i^{(1)} - C_i^{(0)}\right) \qquad (i = 1, \ldots, n) \qquad (4.59)$$

and the species equation becomes

$$\rho^{(0)^2} D^{(1)} \; \frac{d^2 \, C_i^{(1)}}{d \, \eta^2} + \left[\frac{d \, (\rho^{(0)^2} D^{(1)}}{d \, \eta} - \tilde{\delta}^{(1)} \, \rho^{(0)} \, v^{(1)} \right] \frac{d \, C_i^{(1)}}{d \, \eta}$$

$$(i = 1, \ldots, n)$$

$$+ \frac{\tilde{\delta}^{(1)^2}}{\rho^{(0)}} \; \frac{\partial \omega_i^{(0)}}{\partial C_i} \; C_i^{(1)} = \frac{\tilde{\delta}^{(1)^2}}{\rho^{(0)}} \left(\frac{\partial \omega_i^{(0)}}{\partial \, C_i} \; C_i^{(0)} - W_i^{(0)} \right) \qquad (4.60)$$

The scheme for solving the equation above and the thermal equation of state is given in Figure 4.4.

<u>NUMERICAL SOLUTION OF THE EQUATIONS</u>

As written in the previous section, the conservation equations (Eqs. 4.57, 4.58, and 4.60) are of the form

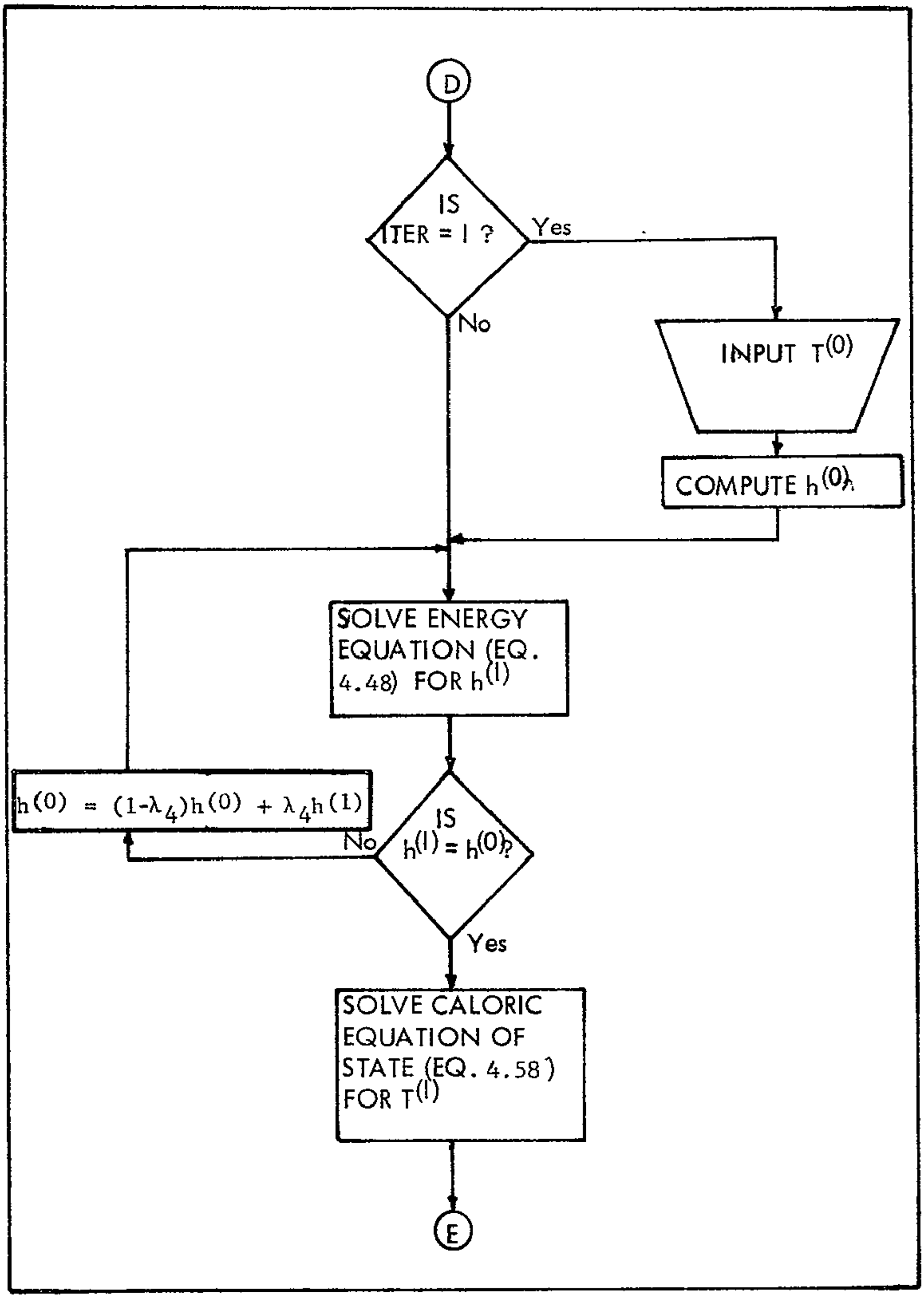


Figure 4.3 Solution of Energy Equation

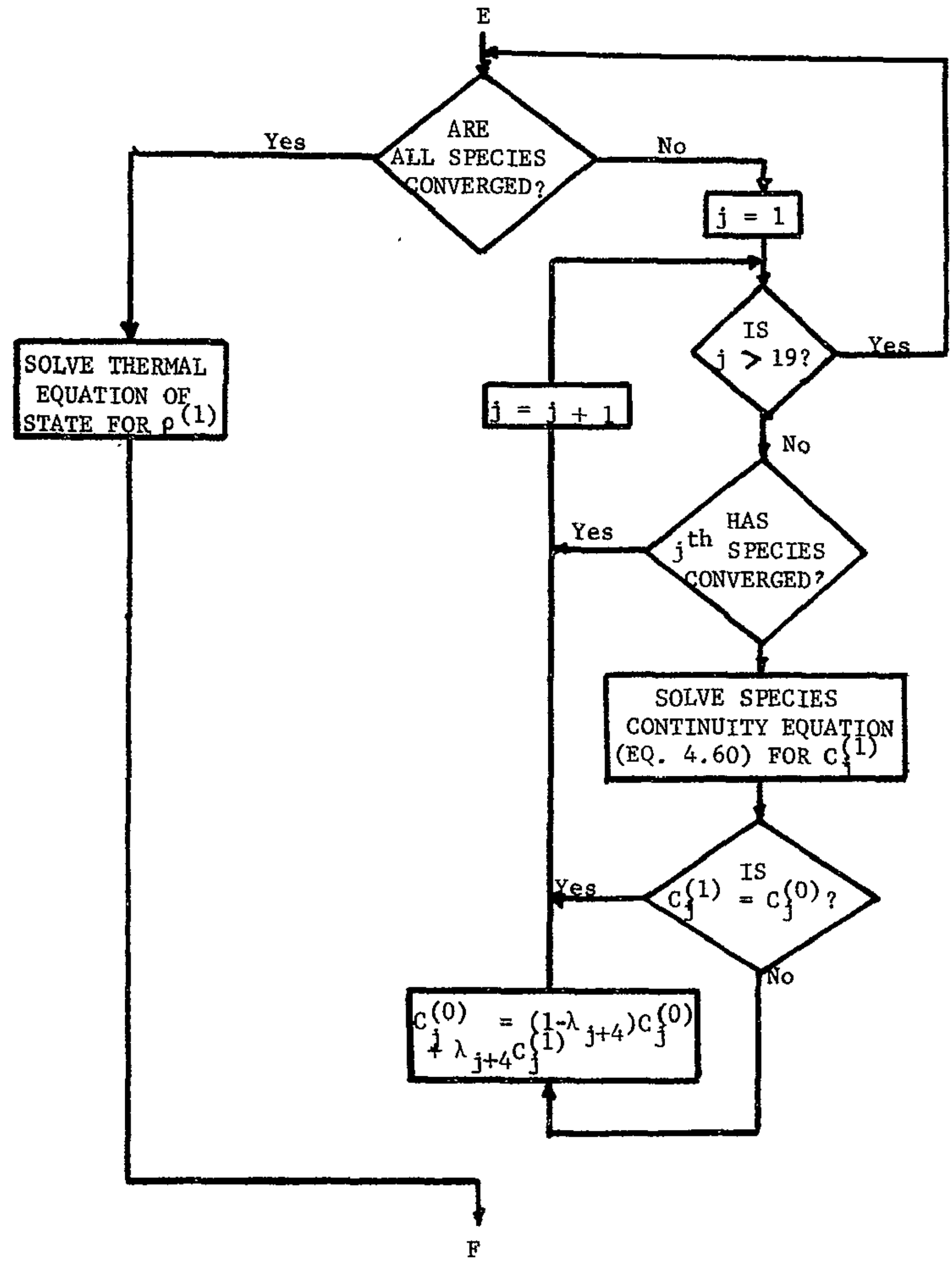


Figure 4.4 Solution of Continuity Equation and Thermal Equation of State

$$\alpha_1 \frac{d^2W}{d\eta^2} + \alpha_2 \frac{dW}{d\eta} + \alpha_3 W = \alpha_4 \tag{4.61}$$

where α_1, α_2, α_3, and α_4 are functions of η. The numerical solution of Eq. 4.61 was accomplished by substituting finite-difference approximations for the derivatives in the equation. These approximations were obtained as follows: Knowing the value of W and its derivatives at any point η_j in the flow-field it is possible to write the Taylor series expression for the value of W at point η_{j-1}

$$W_{j-1} = W_j + \frac{dW_j}{d\eta} (\eta_{j-1}-\eta_j) + \frac{d^2W_j}{d\eta^2} \frac{(\eta_{j-1}-\eta_j)^2}{2} + 0\ (\eta_{j-1} - \eta_j)^3 \tag{4.62}$$

where $W_j = W(\eta_j)$. Knowing W_j and its derivatives it is also possible to write an expression for W_{j+1}

$$W_{j+1} = W_j + \frac{dW_j}{d\eta} (\eta_{j+1}-\eta_j)^2 + 0\ (\eta_{j+1}-\eta_j)^3 \tag{4.63}$$

Neglecting higher order terms and solving Eqs. 4.62 and 4.63 first for $dW_j/d\eta$ and then for $d^2W_j/d\eta^2$ yields

$$\frac{dW_j}{d\eta} \approx \frac{\Delta\eta_{j-1}}{\Delta\eta_j\ (\Delta\eta_j + \Delta\eta_{j-1})} W_{j+1} + \frac{(\Delta\eta_j - \Delta\eta_{j-1})}{\Delta\eta_j\ \Delta\eta_{j-1}} W_j + \frac{(-\Delta\eta_j)}{\Delta\eta_{j-1}\ (\Delta\eta_j + \Delta\eta_{j-1})} W_{j-1} \tag{4.64}$$

$$\frac{d^2W_j}{d\eta^2} \approx \frac{2}{\Delta\eta_j \ (\Delta\eta_j + \Delta\eta_{j-1})} \quad W_{j+1} \quad + \frac{(-2)}{\Delta\eta_j \ \Delta\eta_{j-1}} \quad W_j$$

$$+ \quad \frac{2}{\Delta\eta_{j-1} \ (\Delta\eta_j + \Delta\eta_{j-1})} \quad W_{j-1} \tag{4.65}$$

where $\Delta\eta_j = \eta_{j+1} - \eta_j$.

When Eq. 4.61 is evaluated at η_j and Eqs. 4.64 and 4.65 are introduced the result is

$$\left[\frac{2\alpha_{1,j} - \Delta\eta_j \ \alpha_{z,j}}{\Delta\eta_{j-1} \ (\Delta\eta_j + \Delta\eta_{j-1})}\right] \quad W_{j-1} \ + \left[\frac{-2\alpha_{1,j} + (\Delta\eta_j - \Delta\eta_{j-1}) \ \alpha_{2,j}}{\Delta\eta_j \ \ \Delta\eta_{j-1}}\right.$$

$$\left. + \ \alpha_{3,j}\right] \quad W_j$$

$$+\left[\frac{2\alpha_{1,j} + \Delta\eta_{j-1} \ \alpha_{2,j}}{\Delta\eta_j \ (\Delta\eta_j + \Delta\eta_{j-1})}\right] \quad W_{j+1} = \alpha_{4,j} \tag{4.66}$$

Equation 4.66 is of the form

$$A_jW_{j-1} + B_jW_j + C_jW_{j+1} = D_j \tag{4.67}$$

Letting $\eta_o = 0$ and $\eta_m = 1$ this means that W_o and W_m are known since they are boundary conditions, when Eq. 4.67 is written for $1 \leq j \leq (m-1)$ the resulting set of equations is given in matrix notation by

$$\begin{bmatrix} B_1 & C_1 & & & & \\ A_2 & B_2 & C_2 & & & \\ & A_3 & B_3 & C_2 & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & Am-2 & Bm-2 & Cm-2 \\ & & & & Am-1 & Bm-1 \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \cdot \\ \cdot \\ \cdot \\ Wm-2 \\ Wm-1 \end{bmatrix} = \begin{bmatrix} D_1 - A_1 W_o \\ D_2 \\ D_3 \\ \cdot \\ \cdot \\ \cdot \\ Dm-2 \\ Dm-1 - Cm-1 \quad Wm \end{bmatrix} \tag{4.68}$$

The Thomas algorithm for tridiagonal matrices (Ref. 4.22) is a very efficient scheme for solving sets of linear algebraic equations involving tridiagonal matrices and has been used in this work to solve Eq. 4.68.

The solution of Equations (4.68) was accomplished by writing a computer program to perform the necessary calculations. This program (SLAC- Stagnation Line Analysis with Chemistry) includes the thermodynamic, transport, kinetic, and radiative properties discussed in Chapter 3, as well as, a subroutine to calculate local chemical equilibrium. The equilibrium calculation uses a free energy minimization technique developed by Del Valle and Pike (Ref. 4.23). SLAC is discussed in Appendix A and listed in Appendix B.

This program is a tool which models the stagnation region of an ablator-protected entry vehicle. Adequate boundary-conditions to describe the bow-shock and ablating surface, as described herein, must be specified to utilize this program.

REFERENCES

4.1 Curtiss, C.F. and J.O. Hirschfilder, "Integration of Stiff Equations," Proc. Nat. Acad. Sci, Vol. 38, pp 235-243. 1952.

4.2 Emanuel, George, "Problems Underlying the Numerical Integration of the Chemical and Vibrational Rate Equations in a Near-Equilibrium Flow," Arnold Engineering Development Center, Technical Documentary Report No. AEDC-TDR-63-82, March 1963.

4.3 Moretti, Gino, "A New Technique for the Numerical Analysis of Non-Equilibrium Flows," AIAA Journal, Vol. 3, No. 2, February 1965, pp. 223-229.

4.4 Ferri, Antonio, Gino Moretti, and Simon Slutsky, "Mixing Processes in Supersonic Combustion," J. Soc. Indust. Appl. Math., Vol. 13, No. 1, March 1965, pp. 229-258.

4.5 Treanor, Charles E., "A Method for the Numerical Integration of Coupled First-Order Differential Equations with Greatly Different Time Constants," Math. Comp., Vol. 20, No. 93, January 1966, pp. 39-45.

4.6 Lomax, Harvard, and Harry E. Bailey, "A Critical Analysis of Various Numerical Integration Methods for Computing the Flow of a Gas in Chemical Nonequilibrium," NASA TN D-4109, August 1967.

4.7 Fowler, M.E. and R.M. Warten, "A Numerical Integration Technique for Ordinary Differential Equations with Widely Separated Eigenvalues," IBM Journal, September 1967, pp. 537-543.

4.8 Gear, C.W., "The Automatic Integration of Stiff Ordinary Differential Equations," Proceedings of IFIP Congress, Supplement Booklet, pp. A81-A85 (1968).

4.9 Lomax, Harvard, "On the Construction of Highly Stable, Explicit, Numerical Methods for Integrating Coupled Ordinary Differential Equations with Parasitic Eigenvalues," NASA TN D-4547, April 1968.

4.10 Lomax, Harvard, "Stable Implicit and Explicit Numerical Methods for Integrating Quasi-Linear Differential Equations with Parasitic - Stiff and Parasitic - Saddle Eigenvalues," NASA TN D-4703, July 1968.

4.11 Lomax, Harvard, Harry E. Bailey, and Franklin B. Fuller, "On Some Numerical Difficulties in Integrating the Equations for One-Dimensional Non-equilibrium Nozzle Flow," NASA TN D-5176, April 1969.

4.12 Bailey, Harry E., "Numerical Integration of the Equations Governing the One-Dimensional Flow of a Chemically Reactive Gas," The Physics of Fluids, Vol. 12, No. 11, November 1969, pp. 2292 - 2300.

4.13 Liniger, Werner, and Ralph A. Willoughby, "Efficient Integration Methods for Stiff Systems of Ordinary Differential Equations," SIAM J. Numer. Anal., Vol. 7, No. 1, March 1970, pp. 47-66.

4.14 Seinfelt, John H., Leon Lapidus, and Myungkyu Hwang, "Review of Numerical Integration Techniques for Stiff Ordinary Differential Equations," Ind. Eng. Chem. Fundam., Vol. 9, No. 2, 1970, pp. 266-275.

4.15 Magnus, D.E., and H.S. Schechter, "Analysis of Error Growth and Stability for the Numerical Integration of the Equations of Chemical Kinetics," General Applied Science Laboratories, Inc. Westbury, L.I., New York, Technical Report No. 607, June 1966.

4.16 Fay, James A., and Harvey Kay, "A Finite-Difference Solution of Similar Nonequilibrium Boundary Layers," AIAA Journal, Vol. 5, No. 11, November 1967, pp. 1949-1954.

4.17 Blottner, F.G., "Viscous Shock Layer at the Stagnation Point with

Nonequilibrium Air Chemistry," AIAA Journal, Vol. 7, No. 12, December 1969, pp. 2281-2288.

4.18 Adams, J.C., C.H. Lewis, H.S. Brahinsky, et al., "Effects of Chemical Nonequilibrium, Mass Transfer, and Viscous Interaction on Spherically Blunted Cones at Hypersonic Conditions," Arnold Engineering Development Center, AEDC-TR-69-237, January 1970.

4.19 Liu, Tsog-mou, Hydrogen Injected into Air at an Axisymmetric Stagnation Point, Ph.D. Dissertation, University of California San Diego, California 1970.

4.20 Bellman, R.E., and R.E. Kalaba, Quasilinearization and Nonlinear Boundary Value Problems, American Elsevier Publishing Company, Inc., New York, 1965.

4.21 Radbill, J.R., and G.A. McCue, Quasilinearization and Nonlinear Problems in Fluid and Orbital Mechanics, Americal Elsevier Publishing Company, Inc., New York, 1970.

4.22 Von Rosenberg, D.U., Methods for the Numerical Solution of Partial Differential Equations, American Elsevier Publishing Company, Inc., New York, 1969, p. 113.

4.23 Del Valle, E.G., and R.W. Pike, "Computation of the Equlibrium Composition of Reacting, Gas-Solid Mixtures with Material and Energy Balance Constraints," NASA-RFL-10, March 1970, Reacting Fluids Laboratory, Louisiana State University, Baton Rouge, LA.

CHAPTER 5

RESULTS OF STAGNATION REGION HEATING ANALYSIS

The SLAC program was used to compute the finite-rate and equilibrium-chemistry stagnation-line heating rate of a 9 foot entry vehicle moving at 50,000 feet/second when the free-stream air density is 8.85 $\times 10^{-8}$ slugs/ft^3 and the mass injection rate $(\rho_w v_w/\rho_\infty U_\infty)$ equals .05. The vehicle is protected by a phenolic-nylon ablator with the following elemental composition (elemental mass fraction) : 73.03% carbon, 7.29% hydrogen, 4.96% nitrogen and 14.72% oxygen. The char surface was assumed to be at the sublimation temperature of carbon at .1 atmosphere pressure (Tw = 3,450°K). The shock was assumed to be concentric $(\frac{d\varepsilon}{dx} = 0 , \frac{d\phi}{dx} = 1)$.

These flight conditions and the rate of mass injection were used because they correspond to the conditions obtained by Engel (Ref. 5.1) when the ablator and the equilibrium shock layer solutions are coupled. In other words, according to his results, a vehicle moving under the flight conditions listed above and protected by phenolic-nylon ablates at a 5% rate. Furthermore, Esch (Ref. 5.2) also obtained chemical equibibrium results for the same flight conditions and injection rate. This means that the equilibrium results obtained during the present work may be compared to those obtained by the two investigators mentioned above. The comparison is more valid than in most cases since the same routine was used by all the investigators to solve the Rankine-Hugoniot equations. This means that the shock boundary conditions used were identical. In addition, the radiation model used was the same.

A complete solution to this entry-heating problem would admit non-equilibrium chemistry effects in the ablator response analysis. Such considerations could give different ablator species and (outside) wall temperatures. Even if the ablator effluent into the shock layer were in equilibrium, an entire new series of calculations similar to the ones done by Engel (Ref. 5.1) should be performed to determine the non-equilibrium coupled solution. Ablator non-equilibrium is not in the scope of this research, and non-equilibrium coupling must await the development of an adequate non-equilibrium shock layer analysis. This later development is precisely the subject of this research.

The wall and shock boundary conditions used in this study are given in Table 5.1. Table 5.2 presents a summary of the flight conditions used and the shock layer properties obtained. Since changes in flight conditions strongly affect the behavior of both chemical kinetics and shock layer calculations, the model and the solution scheme developed in this work should be studied over a wide range of flight conditions to determine its adequacy. In the present work only one set of flight conditions was considered, but it is believed that since the conditions are typical of those encountered during atmospheric entry, the model and its solution should work for a range of conditions similar to the ones considered.

COMPUTATION OF BODY HEATING RATE

Since computation of body heating rates for equilibrium and finite-rate chemistry was one of the stated objectives of this investigation, it is proper to discuss how this computation is performed.

As was discussed in Chapter 2, the stagnation line energy boundary condition at the body surface is given in dimensional form by

TABLE 5.1 WALL AND SHOCK BOUNDARY CONDITIONS

Wall $(y = 0)$	Shock $(y = \delta)$
1. $\rho v = 2.23 \times 10^{-4} \ \frac{\text{slug}}{\text{ft}^2\text{-sec}}$	1. $\rho v = -4.45 \times 10^{-4} \ \frac{\text{slug}}{\text{ft}^2\text{-sec}}$
2. $\frac{d(\rho v)}{d\,y} = 0$	2. $\frac{d(\rho v)}{d\,y} = 8.5 \times 10^{-2} \ \frac{\text{slug}}{\text{ft}^3\text{-sec}}$
3. $T = 3{,}450°K$	3. $T = 13{,}000°K$
4. C_i = Chemical equilibrium composition of ablation products at 3,450°K and 0.1 atmosphere	4. C_i = Chemical equilibrium composition of air at 13,000°K and 0.1 atmosphere

TABLE 5.2 SUMMARY OF FLIGHT CONDITIONS AND SHOCK LAYER PROPERTIES

$\rho_\infty = 8.85 \times 10^{-8}$ slub/ft^3

$U_\infty = 5 \times 10^4$ ft/sec

$\rho_w = 8.44 \times 10^{-6}$ slug/ft^3

$v_w = 2.64 \times 10^1$ ft/sec

$T_w = 3{,}450$°K

$\rho_\delta = 1.7 \times 10^{-6}$ slug/ft^3

$v_\delta = -2.62 \times 10^3$ ft/sec

$T_\delta = 13{,}000$°K

$p_\delta = 0.1$ atm

$\bar{\rho} = 0.0536$

$R = 9$ ft

Injection rate = 5%

$$(\rho vH - k \frac{dT}{dy} + \sum_i^n h_i J_{i,y} + q_{r,y})^+ =$$

$$(\rho vH - k \frac{dT}{dy} + \sum_i^n h_i J_{i,y} + q_{r,y})^- = Q \qquad (2.50e)$$

The heating rate Q is the net amount of energy heating the body. It must be recalled that the + and - superscripts refer to the shock layer and char sides of the interface, respectively. Q is computed by:

$$Q = (\rho vH - k \frac{dT}{dy} + \sum_i^n h_i J_{i,y} + q_{r,y})^+ \qquad (5.1)$$

Preliminary calculations showed the first term in this equation to be negligible when compared to the other terms in the equation, for this reason it was not included in the computation of Q. The heating rate to the body was expressed as

$$Q = Q_c + Q_d + Q_R \qquad (5.2)$$

where

$$Q_c = - k \frac{dT}{dy} = \text{Heat transfer by convection} \qquad (5.3)$$

$$Q_d = - \sum_i^n h_i J_{i,y} = \text{Heat transfer by diffusion} \qquad (5.4)$$

$$Q_R = q_{R,y} = \text{Heat transfer by radiation} \qquad (5.5)$$

FINITE-RATE CHEMISTRY RESULTS

The SLAC program was used to compute non-equilibrium temperature, enthalpy, density, velocity and mass fractions profiles, and the body heating rate for the flight conditions previously discussed. Two sets of runs were performed, one using the chemical kinetics model described in Chapter 3, and the other using a modified kinetics model developed

by Balhoff (Ref. 5.3). The modified model assumes the same reactions used in the model described in Chapter 3, but the coefficients of the forward reaction rate constants were modified as shown in Table 5.3. Comparison of these coefficients to those in Table 3.8 shows that while the reactions used are the same, Balhoff's chemistry model uses significantly smaller reaction rate constants in 9 of the 16 reactions considered. Balhoff's model is in effect a modification of the original model used since it was found out that the frequency factors computed using collision theory were in error.

The results from both runs are presented here (Table 5.4) to give an indication of how sensitive the heating rate is to the value of the rate constants used. It must be pointed out that the solution using the rates in Table 5.3 were obtained in about 5% of the computer time, it took to obtain the solution with the original chemistry model (18 hours vs. 45 minutes in an IBM 360/65). This was probably due to the fact that the rates used in the original chemistry model were much faster than those in the modified model, and this made convergence of the species solution more difficult.

CHEMICAL EQUILIBRIUM RESULTS

Results obtained from the SLAC program using an equilibrium chemistry model for the flight conditions previously discussed provide profiles of temperature, enthalpy, density, velocity and mass fractions from the body surface ($y/\delta = 0$) to the shock ($y/\delta = 1$), and the body heating rate. It was maintained above that one of the reasons for choosing the flight conditions studied was the fact that both Engel (Ref. 5.1) and Esch (Ref. 5.2) had studied the same condition and, therefore, the validity of the equilibrium results obtained from SLAC

TABLE 5.3 MODIFIED COEFFICIENTS OF THE FORWARD REACTION RATE CONSTANTS

j^{th} Reaction	a_{tj}	b_{fj}	e_{fj}	Comments
1	1.09 E 10	0.5	71,000	*
2	4.50 E 11	0.5	35,000	
3	1.10 E 21	-1.5	224,900	
4	3.60 E 18	-0.82	103,000	
5	2.80 E 12	0.5	313,000	
6	2.90 E 12	0.5	333,000	
7	2.20 E 20	-1.0	131,800	
8	1.09 E 10	0.5	120,000	*
9	1.22 E 10	0.5	140,000	*
10	1.23 E 10	0.5	117,000	*
11	8.50 E 19	-1.0	257,900	
12	1.28 E 10	0.5	190,000	*
13	1.20 E 10	0.5	155,000	*
14	1.21 E 10	0.5	165,000	*
15	1.09 E 10	0.5	115,000	*
16	1.09 E 10	0.5	145,000	*

*This rate constant is different from that used in the model described in Chapter 3.

TABLE 5.4 NON-EQUILIBRIUM HEATING RATES AS PREDICTED BY TWO SETS OF RATE CONSTANTS

Rate Constants	# Q_c	# Q_d	# Q_R	# Q
Table 5.3	30	1	138	169
Table 3.8	4	1	106	111

watts/cm^2

could be corroborated.

Before proceeding to discuss and compare the results obtained to those obtained by Engel (Ref. 5.1) and Esch (Ref. 5.2), it is relevant to discuss the differences in the models used. Firstly, Engel used transport properties of air thoughout the flow field (including the region from the stagnation point to the body) while Esch and the present work used properties of air and ablation products throughout the flow field. Secondly, the species wall boundary conditions are given by (See Eq. 2.50b).

$$(\rho v)_w \, C_{i,w} - J_{i,w} = (\rho v)_w \, C_i^- \quad (i = 1, \ldots, \eta) \tag{5.6}$$

where the term J_i^-, which corresponds to mass diffusion inside the char has been neglected, and the C_i^- are the mass fractions obtained from the ablator response analysis. Esch assumed that mass diffusion could be described by using a binary diffusion coefficient ($J_i = -\rho D \frac{dC_i}{dy}$), and used boundary conditions of the third kind:

$$(\rho v)_w \, C_{i,w} + \left(D \frac{dC_i}{dy}\right)_w = (\rho v)_w \, C_i^- \quad (i = 1, \ldots, \eta) \tag{5.7}$$

Engel assumed there was no diffusion throughout the flow field ($J_i(y) = J_{i,w} = 0$), and used boundary conditions of the first kind:

$$C_{i,w} = C_i^- \quad (i = 1, \ldots, \eta) \tag{5.8}$$

In the present investigation mass diffusion was allowed throughout the shock layer but, in order to simplify the solution to the species equations, the boundary conditions used at the wall were of the first kind:

$$C_{i,w} = C_i^- \quad (i = 1, \ldots, n) \tag{5.9}$$

Lastly, both Engel and Esch solved the energy equation in temperature form while in the present work an enthalpy form of the energy equation was used. As was discussed in Chapter 4, many attempts were made at solving the energy equation in temperature term, however, it was not possible to obtain a solution, and therefore, an enthalpy form of the equation was used.

The chemical equilibrium heating rates obtained by Engel, Esch and the present investigation are compared in Table 5.5. It must be noticed that Engel and Esch did not include the heat transfer to the body by diffusion.

The significant difference in convective heating rates, with the present investigation yielding an essentially negligible heating rate by convection, results because the present investigation predicts smaller temperature gradients near the wall than those predicted by Engel and Esch. In other words, the present investigation predicts a higher convective heating blockage effect resulting from injection of ablation products into the flow field.

COMPARISON OF PREDICTED NON-EQUILIBRIUM AND EQUILIBRIUM HEATING RATES

From the results shown in Tables 5.4 and 5.5 it is evident that the predicted equilibrium heating rate is approximately three times as large as the non-equilibrium heating rate, and for both cases most of the body heating is the result of radiation.

The lower radiative heating rate predicted by the non-equilibrium computation may be explained by examining the changes in chemical composition which occur in the shock layer. When the chemical compositions predicted by the equilibrium and the non-equilibrium calculations are

TABLE 5.5 EQUILIBRIUM HEATING RATES OBTAINED BY DIFFERENT INVESTIGATORS

INVESTIGATOR	$Q_c^{\#}$	$Q_d^{\#}$	$Q_R^{\#}$	$Q^{\#}$
Engel (Ref. 5.1)	50	¢	350	400
Esch (Ref. 5.2)	120	¢	397	517
Present Investigation	3	1	388	392

¢ Did not consider heat transfer by diffusion

watts/cm^2

compared, it becomes evident that they are markedly different and that finite-rate chemistry effects are present throughout the flow-field (The complete set of dimensionless temperature, enthalpy, density, velocity and mass fraction profiles for the two non-equilibrium and the equilibrium runs are presented in Appendix C. Those profiles needed in the following discussion will be reproduced in this chapter.)

An examination of the chemical composition of equilibrium ablation products shows that as they leave the char surface and are subjected to rapidly increasing temperatures, the decomposition process begun in the char continues as the "large" molecules, C_3H, C_2H, C_4H, HCN, C_2H_2, and H_2 are broken down (for example see Fig. 5.1) into simpler species such as C_3, C_2, CO, CN, N_2, C and H (see Fig. 5.2). As the ablation products flow towards the stagnation point, this process continues as C_3, C_2, CO, CN, and N_2 are converted to C, C^+, N, and O (see Figs. 5.2 and 5.3).

Near the stagnation point, most of the carbon is ionized to C^+ and, due to diffusion of ablation products past the stagnation point, the concentration of C^+ persists for some distance past the stagnation point before decreasing to zero (see Figs. 5.3 and 5.4).

In the region from the char surface to the stagnation point the non-equilibrium results predict that the ablation products will decompose at a much slower rate, when compared to the rate predicted by the equilibrium results. The "large" molecules (C_3 H, C_2 H, C_4 H, HCN, C_2H_2, and H_2) and others (C_3, CO, CN, and N_2) begin to break down into C, C^+, C_2, N, N^+, O^+, and H (see Figs. 5.1, - 5.3). But the ablation products reach the stagnation point before this process is completed and the results are much lower concentrations of C, C^+, N, O, and H in

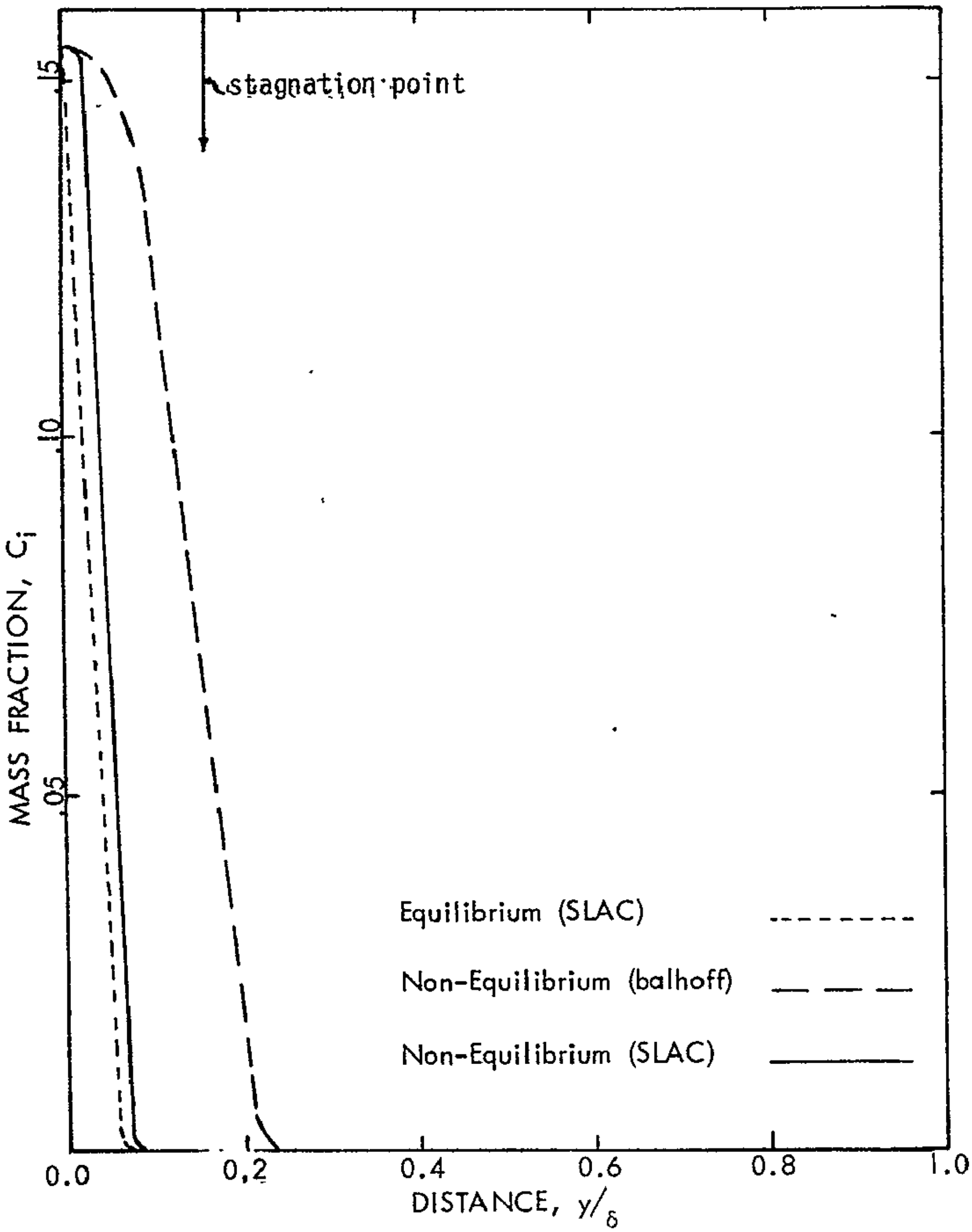


Figure 5.1 Mass Fraction Profiles for C_3H

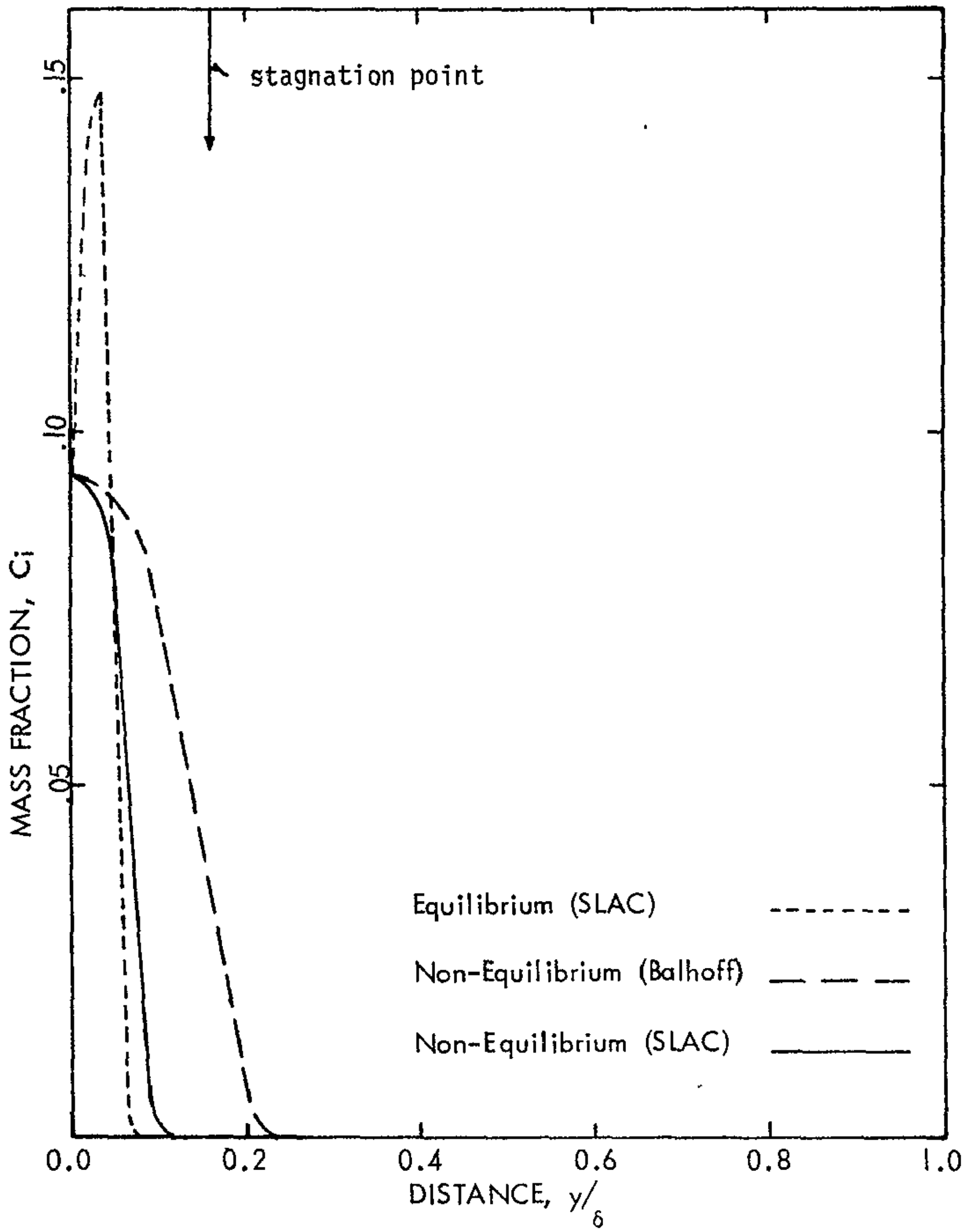


Figure 5.2 Mass Fraction Profiles for C_3

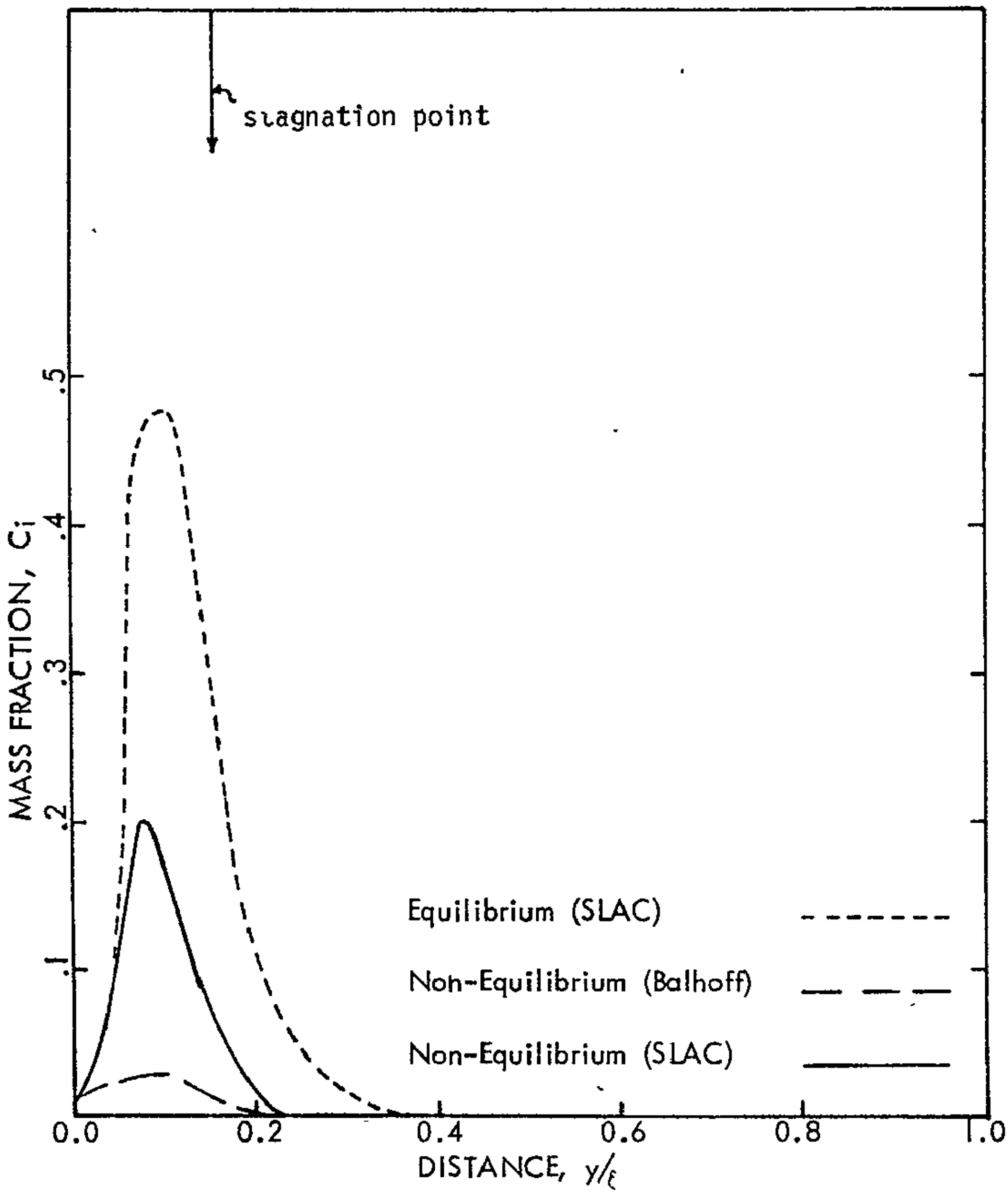


Figure 5.3 Mass Fraction Profiles for C

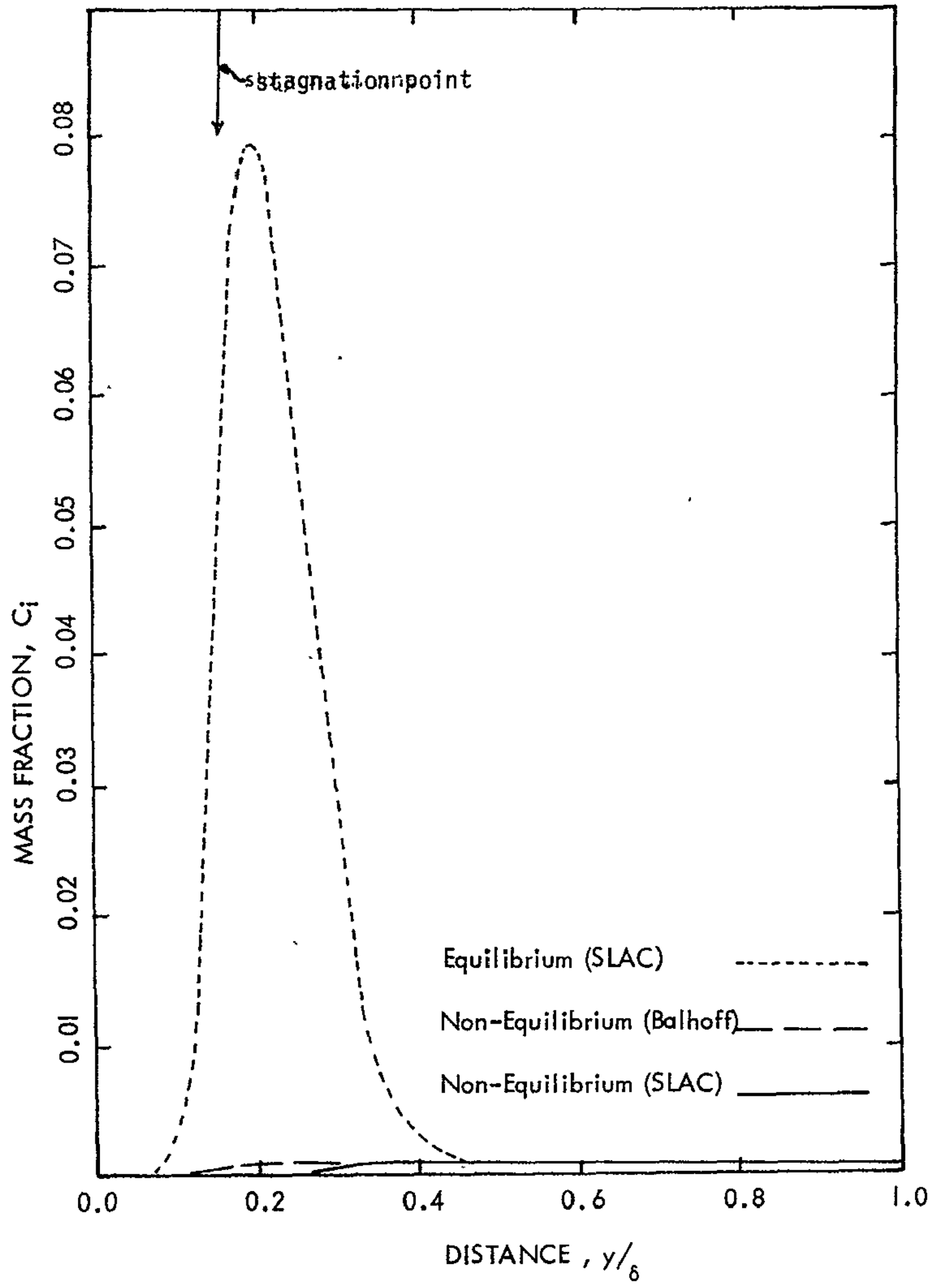


Figure 5.4 Mass Fraction Profiles for C^+

this region (see Fig. 5.4).

In the shock layer region of the flow-field, equilibrium results predict that as the air components are cooled down a process of rapid de-ionization occurs with most of the N^+ and O^+ being converted to N and O (see Figs. 5.5 and 5.6). On the other hand, non-equilibrium results predict that the air components will flow from the shock to the stagnation point without undergoing any changes until very close to the stagnation point where their concentrations begin to decrease and fall rapidly past the stagnation point. As a result of these "frozen" concentrations, the region between the shock and the stagnation point has much lower concentrations of N and O (and much higher concentrations of N^+ and O^+) than those predicted by the equilibrium analysis (see Figs. 5.5 and 5.6).

Therefore, the non-equilibrium analysis predicts much lower concentrations of C, CN, C^+, N_2, N, O, and H and much higher concnetrations of C_2, CO, C_3H, C_2H, C_4H, HCN, C_2H_2, N^+, O^+, H_2, and e^- than the equilibrium analysis. Since C, N, O, and H line and continuum mechanisms are the major contributors to radiative energy transport in the shock layer (Ref. 5.4), under non-equilibrium chemistry conditions the resulting radiative heat transfer to the body is significantly lower than the one predicted by chemical equilibrium analyses.

DIFFICULTIES ASSOCIATED WITH THE SOLUTION

This research demonstrates that the speed of obtaining a solution, if one can be obtained, depends on a large number of factors including: the formulation of the model, how realistic the values used for properties are, the numerical techniques used, and how such techniques are implemented.

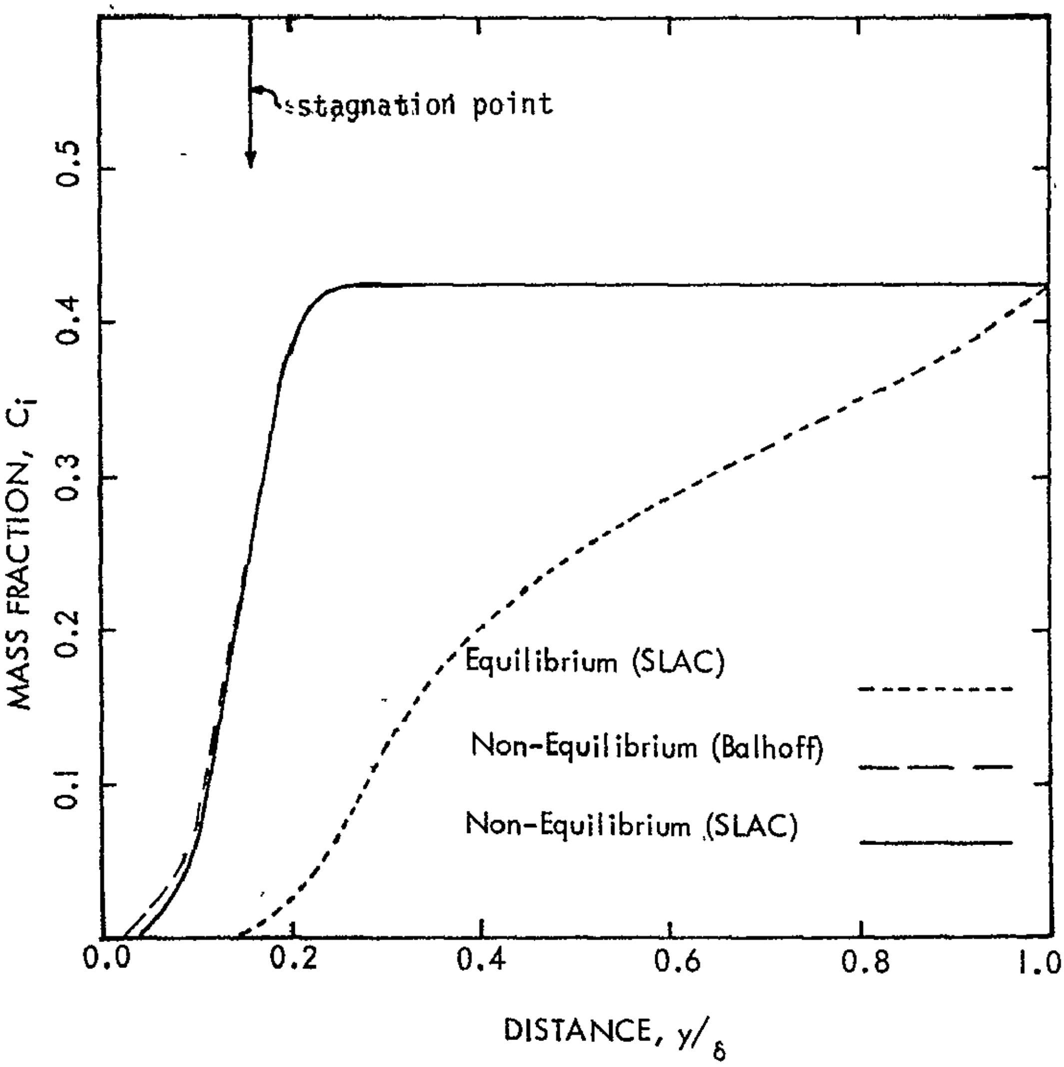


Figure 5.5 Mass Fraction Profiles for N^+

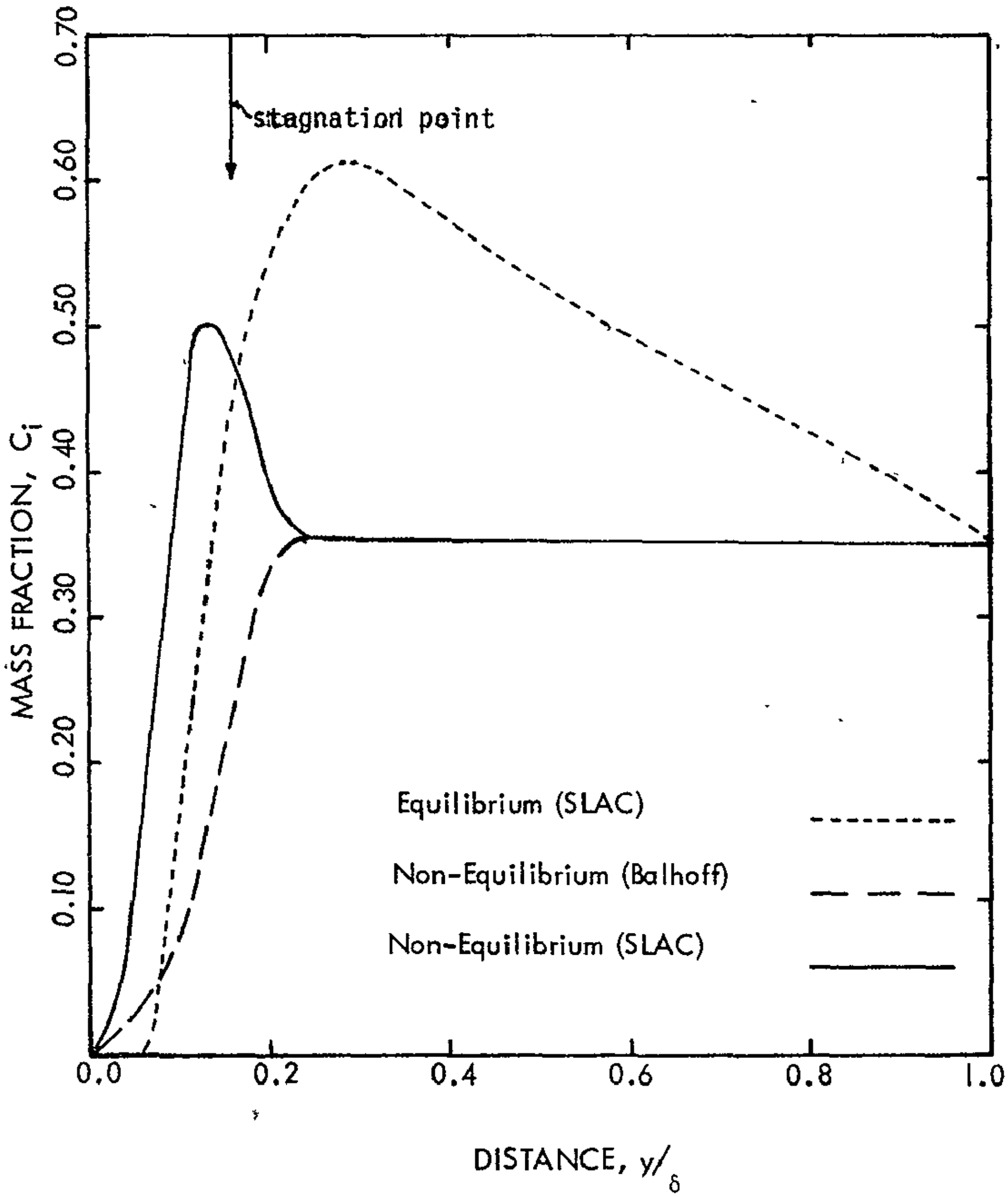


Figure 5.6 Mass Fraction Profiles for N

The attempts at using the energy equation in terms of temperature failed because the temperature profile is very sensitive to the values of the other dependent variables and the energy equation in this form is apparently more strongly coupled to the species equations. The enthalpy formulation, on the other hand, permits uncoupling of the equations and makes it possible to obtain a converged solution.

However, although a solution was determined using the energy equation in terms of enthalpy, obtaining such a solution was extremely difficult. Much of this difficulty arose because of the use of unrealistically high reaction rates. The non-equilibrium solution using the original chemical kinetics model took about 18 hours of time in an IBM 360/65 computer. The solution using the modified kinetics model took about 45 minutes in the same computer. This means that the unrealistic rates increased the difficulty in obtaining a solution.

As was discussed in Chapter 4, a number of numerical techniques have been demonstrated to be inappropriate for solving chemical kinetics problems. However, even those techniques which may be valid will not work unless they are implemented properly. For example, the scheme used to uncouple and iterate on the equations will, in many cases, make the difference between success and failure. Also, the weighting factors used to obtain new guesses from the old and new solutions (the λ's in Figs. 4.1 - 4.4) determine if the solution will converge. Unfortunately, there is no straightforward procedure for determining how to implement a technique properly; a great deal of experimentation is required before success is attained.

REFERENCES

5.1 Engel, Carl D., Ablation and Radiation Coupled Viscous Hypersonic Shock Layers, PhD. Dissertation, Louisiana State University, Baton Rouge, Louisiana (1971)

5.2 Esch, Donald D., Stagnation Region Heating of a Phenolic-Nylon Ablator During Return from Planetary Missions, Ph.D. Dissertation Louisiana State University, Baton Rouge, Louisiana (1971).

5.3 Balhoff, John F., Research Associate, Chemical Engineering Department, Louisiana State University, Private Communication.

5.4 Engel, op, cit., Vol. I, p. 126

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

CONCLUSIONS

From the results obtained in the present investigation the following conclusions can be derived:

A finite-rate stagnation-line shock layer solution which contains a reasonable kinetics model to describe atmospheric entries protected by phenolic-nylon ablators was developed.

The model was used to determine the non-equilibrium shock layer structure and body heating rates under flight conditions typical of return from planetary missions. Although only one set of conditions was considered, it is believed that the method of solution is adequate for a range of conditions similar to the ones considered.

2. Finite-rate chemistry effects are significant for the flight conditions considered.

The results obtained show that as the ablation products enter the shock layer they react much slower than under equilibrium conditions. The air components entering through the shock are not de-ionized, as the equilibrium analysis predicts, but remain "frozen" throughout most of the shock region. The validity of these results are, of course, dependent upon the accuracy of the chemical kinetics model used. The degree of confidence in the results is high because: the kinetics model used is

the most complete that has been used in this type of application, the reaction rates used were the best rates available, and for both sets of reaction rates (the original and the modified) the same finite-rate chemistry effects were present.

3. For the flight conditions studied the total heating rate to the body is significantly lower under non-equilibrium than under equilibrium chemistry conditons.

The predicted non-equilibrium heating rate is approximately one third as large as the predicted equilibrium heating rate. This results from the fact that the non-equilibrium radiative heating rate is approximately one third as large as the equilibrium radiative heating, and in both instances, most of the total heating results from radiation. The significantly lower radiative heating rate is a direct result of finite rate chemistry effects which result in much lower concentrations of C, N, O, and H, the species responsible for most of the line and continuum radiative transfer.

RECOMMENDATIONS

Considering the conclusions presented above, it is recommended that:

1. The mathematical model for the stagnation-line analysis of an entry vehicle presented herein should be studied further to determine if its computational speed can be increased.

The present version of SLAC should be developed from the research tool it is today until it becomes an effective engineering tool. The principal obstacle that needs to be overcome is the amount of computer time required to obtain a solution. The computational speed is a function of: the model formulation, how realistic the properties used are,

the schemes for uncoupling and iterating on the equations, and the numerical techniques used. Much improvement in computational speed was obtained by considering different variations on all these factors. It is believed that the computational speed of the present solution could be increased by at least a factor of 2 by further consideration of uncoupling and iterating schemes.

2. Studies of finite-rate chemistry effects be carried out under flight conditions different from those considered in the present work.

The objectives of the proposed studies should be to determine: a) the range of flight conditions over which solutions may be obtained using the implemented model; and b) the range of conditions over which finite-rate chemistry effects are significant and how they affect the heating rates to the body.

3. Research be carried out on the effect of assuming more realistic shock and wall boundary conditions.

This investigation should consider using non-equilibrium compositions of air at the shock as boundary conditions and using boundary conditions of the third kind at the wall. The use of non-equilibrium compositions of air will result in increased concentrations of N and O and less N^+ and O^+ when compared to the equilibrium boundary conditions. This might result in increased radiative heating rates since N and O are optically more active than N^+ and O^+. By perturbing the shock boundary conditions used in SLAC an estimate of the importance of this effect could be developed. In the present investigation the mass and energy boundary conditions used at the wall were of the first kind, however the mass and energy surface balances yield conditions of the third kind.

A study should be carried out to determine if using the more complete conditions has any significant effect on the heating rate.

4. Coupled solutions for both non-equilibrium ablation products and shock layers should be determined.

A complete solution to the quasi-steady entry problem requires that the ablator response be coupled to the existing flight conditions through the shock layer. This can be accomplished by performing multiple calculations of ablation and shock layer behaviour and matching conditions at the interface between the ablator surface and the flow field. At present, ablator-shock layer response under chemical equilibrium conditions has been studied, however, additional studies should be carried out coupling the finite-rate solutions determined from SLAC to those obtained from a non-equilibrium ablator-response model.

APPENDIX A

USER'S MANUAL FOR SLAC

INTRODUCTION

This appendix will serve as a user's manual for the SLAC (Stagnation Line Analysis with Chemistry) program. The program implements a model designed to predict the stagnation line viscous, reactive, and radiative coupled shock layer structure, and the resulting heating rates produced by a blunt body during super orbital entry into planetary atmospheres. The problem was formulated in Chapter 1, and the equations describing the flow-field (the stagnation line boundary layer equations (See Table 2.7), and the wall and shock boundary conditions (See Table 2.8), were derived in Chapter 2. The thermodynamic, transport, radiative, and chemical kinetic properties are described in Chapter 3. The program can also be used to compute flows in chemical equilibrium, and for this purpose it utilizes a free energy minimization routine developed by Del Valle and Pike (Ref. A.1). The numerical produres used to solve the model are described in Chapter 4.

The SLAC program results from extensions performed on a program (VISRAD) primarily developed by Engel (Ref. A.2), and in a program (SLAB) developed by Esch (Ref. A.3) from VISRAD. These programs result from the efforts of many individuals over a considerable period of time. SLAC was designed to be used for thermal environment prediction studies. It

represents a significant tool for studying a variety of atmospheric entry heating problems. The program, written in FORTRAN IV, is capable of performing the following types of analyses:

- Stagnation line solutions
- Coupled diffusive, convective and radiative flux calculations
- Emission, and line and continuum radiation calculations
- Binary diffusion calculations
- Finite-rate or equilibrium chemistry calculations

PROGRAM PROCEDURES

The SLAC computer program was developed following a philosophy of minimizing user's effort and maximizing program flexibility and adaptability. Accordingly, the basic program logic as shown in Figure A.1 is quite simple. However these basic subprograms are supported by 22 subroutines and 7 function subprograms. Each of these modules performs computational functions which are of a basic nature (e.g., computation of thermodynamic properties), and allows for modification, substitution, addition or deletion of existing modules with minimum effort.

In order to minimize input requirements, three techniques were used. The first consists of internal initialization of values for temperature, density, viscosity, and stand-off distance which are necessary to start the solution procedure. If better guesses are available they may be input as discussed in the next section. The second technique involves internal specification in BLOCK DATA of problem-defining parameters such as the elemental composition at the surface and thermodynamic curve-fit constants which are changed quite infrequently. The third technique consists of internally selecting program options if an option variable is left blank on an input card. In this procedure the most commonly used options are performed when a blank is input.

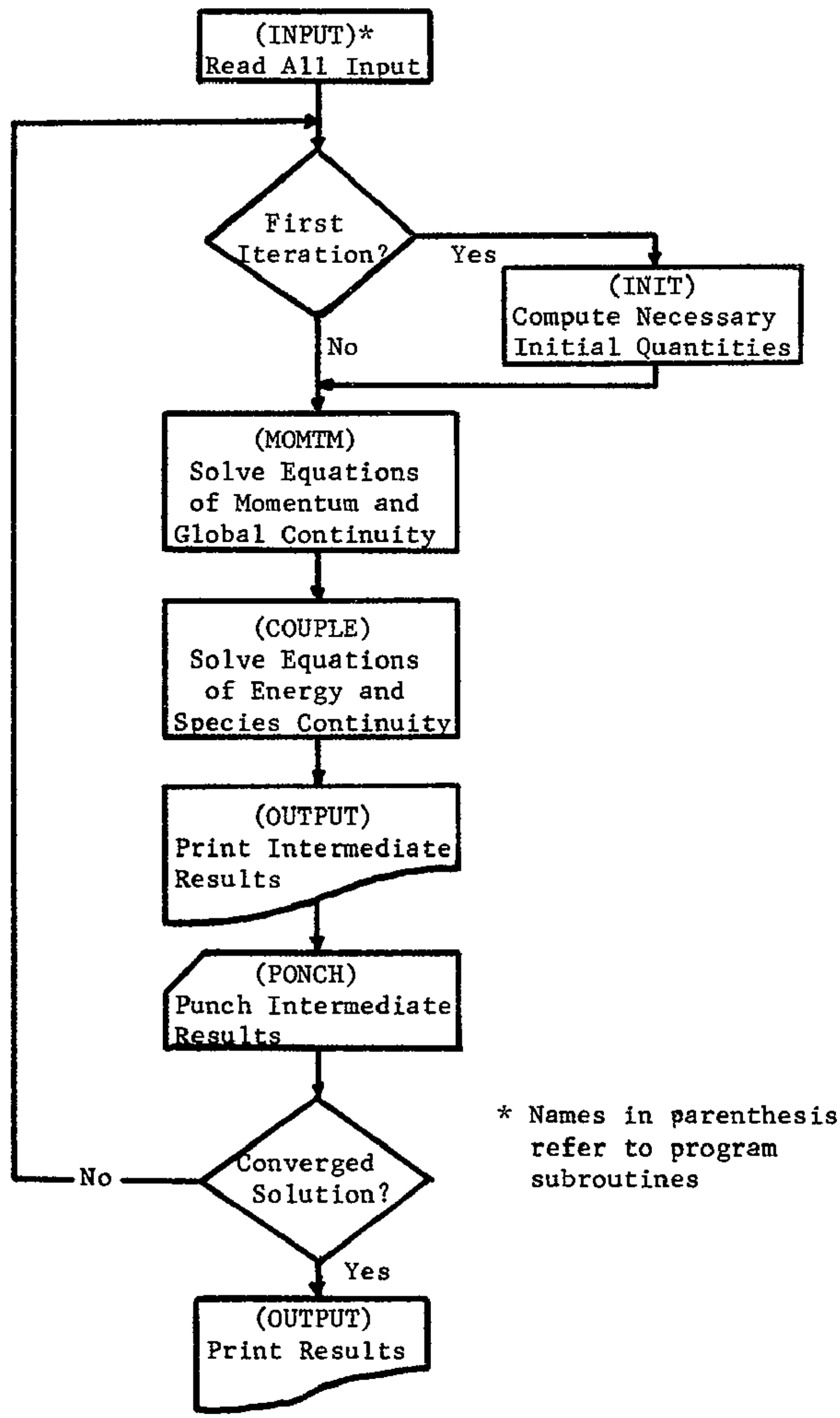


FIGURE A.1 PROGRAM FLOWCHART

The start up of the SLAC program can be achieved in a number of ways. As stated previously, internal guesses are available to begin the iteration procedure. Two types of temperature profiles are available. One for no mass injection and the other for mass injection. These profiles are usually quite satisfactory as initial guesses if an emission radiation coupled problem is to be run. However, if a line and continuum radiation coupled problem which includes mass injection, and finite-rate chemistry is to be run the internal guess may not be accurate enough. Consequently, it might be necessary to input a better temperature profile guess.

INPUT GUIDE

All inputs to the SLAC computer program are read from cards supplied by the user; no tapes are required. The basic inputs consist of parameters defining flight conditions (free-stream velocity and density), blunt body radius, wall temperature and mass injection rate. Additional input parameters are required to determine which program options are desired, and to provide the necessary guesses of dependent variables. After each overall iteration, the program outputs a deck of cards containing the values of the dependent variables and other pertinent parameters in such a manner that a given case run may be interrupted and continued at a later time by using the produced deck of punched cards to restart the run. Multiple case runs, and hence entire trajectories, can be processed by placing the input data for each new case behind the data for the previous one.

Table A.1 presents the card input formats for SLAC and Table A.2 provides a corresponding definition of variables.

In single case runs Card 10 must be followed by a card with the characters END punched in columns 1, 2 and 3 to indicate the run has ended. In multiple case runs the END card should be placed after Card

TABLE A.1 CARD INPUT FOR SLAC

Card	Variables	Format
1	TITLE, IEM	18A4, I8
2	KEEP, NETA, IRAD, ITYPE, MAXM, MAXE MAXD, LT, IPHI, FPRCT, TPRCT, IDEBUG	9I5, 2E12.0, 2X, I1
3	UNIF, RINF, R, TWK, HTØTAL, RVW	6E12.0
4	DELTA, DTIL, RZB, RE, PDTIL	5E12.0
5	T(I)	6E12.0
6A	RHØ(I)	6E12.0
6B	RM(I)	6E12.0
7	DEPS	E12.0
8	ETA(I)	6E12.0
9	NDEBUG, TOL, IAB	I5, 5X E10.4, I5
10	CWALL(J)	5E15.8

TABLE A.2 VARIABLE DEFINITIONS FOR SLAC

Card	Variable	Description
1	TITLE	Title for identification of the problem
	IEM	Number of overall iterations performed. IEM = 0 for case fun initiation, IEM>0 for restarts
2	KEEP	Indicator to determine if the temperature profile from the previous case is to be kept as a guess for the current case. KEEP = 0 Temperature not kept KEEP = 1 Temperature kept
	NETA	The number of points to be used in the shock layer profile. If NETA = 0, a set of 51 equally spaced points will be used. If NETA>0 card 8 must be read.
	IRAD	A variable used to specify the type of solution. IRAD = 1 Convective solution only = 2 Uncoupled radiation = 3 Coupled radiation solution
	ITYPE	A variable used to specify the type of radiation model to be used. ITYPE = 0 Line and continuum radiation model = 1 Emission radiation model
	MAXM	Maximum number of iterations allowed in the internal momentum loop. If MAXM = 0, it is internally set = 15.
	MAXE	Maximum number of iterations allowed in the energy-species continuity equation and in the overall momentum-energy-species loop. If MAXE = 0, it is internally set = 15.
	MAXD	Maximum number of iterations allowed in the external momentum loop. If MAXD = 0, it is internally set = 15.

Card	Variable	Description
	LT	Indicator to determine if a temperature guess and if ρ and ρv- guesses are to be read in. LT = 0 Cards 5 and 6 are not read. = 1 Card 5 but not card 6 is read. = 2 Cards 5 and 6 are read.
	IPHI	Indicator to determine if the shock curvature is to be input. IPHT = 0 $d\varepsilon/d\varepsilon = 0$ is internally set. = 1 Card 7 is required for input.
	FPRCT	Convergence tolerance for each point the f' profile. If FPRCT = 0.0 it is internally set = .005.
	TPRCT	Convergence tolerance for each point in the T profile. If TPRCT = 0.0, it is internally set = .005.
	IDEBUG	A switch to allow intermediate printout to be obtained at each iteration IDEBUG = 0 No print. = 1 Print is given
3	UNIF	The free-stream flight velocity (U_∞) i feet/sec.
	RINF	The free-stream density (ρ_∞) in slugs/ft^3.
	R	Principal body redius in feet.
	TWK	Wall Temperature in degrees Kelvin.
	HTOTAL	Total free-stream enthalpy in ft^2/sec.^{2}. If HTOTAL = 0.0, it is set to $U_\infty^2/2$. (Free-stream static enthalpy is assumed negligible).
	RVW	Mass injection rate $(\rho v)_w/(\rho U)_\infty$.
4	DELTA	An initial guess for the shock standoff distance δ/R. If DELTA = 0.0, a guess is supplied by program.

Card	Variable	Description
	DTIL	A guess for the transformed standoff distance δ/R. The program will also supply this value if DTIL 0.0
	RZB	The density ratio across the shock $\bar{\rho}=\rho_\infty/\rho_\delta$ If RZB is input as 0.0, the code will determine a value.
	RE	The Reynolds number for the problem, $Re_s = U_\infty R\rho_\delta/\mu_\delta$. This quanity is determined by the program if RE is input as 0.0.
	PDTIL	Convergence tolerance placed on δ for total solution convergence. If PDTIL = 0.0, it is internally set = .001.
5	T(I), I = 1, NETA	An initial guess for the dimensionless shock layer temperature profile (T/T_δ). This card is required only if LT>0.
6A	RHØ(I),I=1, NETA	An initial guess for the dimensionless shock layer density profile (ρ/ρ_δ). This card is required only if LT = 2.
6B	RM(I), I=1, NETA	An initial guess for the dimensionless shock layer ρv profile $(\rho v/\rho_\delta\ v_\delta)$. This card is required only if LT = 2.
7	DEPS	The stagnation line shock curvature $(d\varepsilon/dx)$. If IPHI = 0, then $d\varepsilon/dx$ = -.0 is internally set. If IPHI = 1, Card 7 is read and $d\varepsilon/dx$ is supplied by the user.
8	ETA(I), I=1, NETA	The grid shock layer points at which the solution profiles are to be computer. If NETA = 0, $\Delta\eta$ is set to 0.02 and ETA (I) is computed by the program. (ETA(1)=0.0→ wall, ETA (NETA)=1.0→ shock).
9	NDEBUG	Debug option to output thermodynamic curve-fit equations and intermediate results from NDEBUG = 0 No output. = 1 Output given.

Card	Variable	Description
	TOL	Convergence criteria for CHEMEQ. If TOL is input as 0.0, the code will set it to 0.001.
	IAB	A variable used to specify the type of chemistry model to be used. IAB = 0 Finite-rate chemistry; IAB = 1 Equilibrium chemistry
10	CWALL(J), J=1,NSP	Wall mass fractions. NSP = 20

J = 1→O_2	6→$N^{\pm}$	11→CO	16→C_3H
2→N_2	7→E^-	12→C_3	17→C_4H
3→O	8→C	13→CN	18→HCN
4→N	9→H	14→C_2H	19→C_2
5→O^+	10→H_2	15→C_2H_2	20→C^+

10 of the last case to be run.

Some caution should be exercised when preparing an input for this program. The program considers 5 elements, including electrons, and twenty species listed under CWALL (J). The set of species was selected for an air atmosphere and an phenolic-nylon ablator. If another ablator is selected for study and this set of species is appropriate, no alteration of the program is required. All that is required is a card input of wall mass fractions of the ablator selected on Card 10. If extensive study of a different ablator using this program is anticipated, the user may find it convenient to change the wall composition stated in BLOCK DATA under CWALL rather than reading in the data for each run.

If required, a change to another set of species can be made with comparatively little difficulty. Thermodynamic and transport properties may be altered by changing the curve - fit constants in BLOCK DATA. The thermodynamic curve - fit equations were listed in Table 3.1 and the correspondence between the coefficients in the table and those in the program is

For $1000 < T \leq 6000$			$6000 < T^oK$
AI	= A_1	=	AII
BI	= A_2	=	BII
CI	= A_3	=	CII
DI	= A_4	=	DII
EI	= A_5	=	EII
FI	= A_6	=	FII
GI	= A_7	=	GII

where the coefficients are dimensioned to include a value for each species.

The species ordering is given in the SP array with corresponding ordering in SMW (i.e. species molecular weight) array in BLOCK DATA. The transport properties curve-fit equations are listed in Tables 3.3 (viscosity) and 3.4 (thermal conductivity) and the correspondence between the coefficients in those tables and those in the program is:

Viscosity:

Table 3.3		SLAC
a	=	V1
b	=	V2
c	=	V3

Thermal Conductivity

Table 3.4		SLAC
a	=	K1
b	=	K2

If new species and reactions are to be included in the finite-rate calculations, then subroutine FG2 must be modified accordingly. Finally, a study should be made to determine if radiatively important species are to be included.

OUTPUT DESCRIPTION

This section presents a description of the program output and definitions of the output symbols. The program produces both printed output and punched card output. The printed output provides a human-readable record of input parameters, intermediate results and the final solution. The punched card output provides a machine-readable medium for restarting runs.

Printed Output

The first page of output is a print of the input data. This is provided for a check of the input and an identification of the problem.

All quantities on this page are defined in the Input Guide section. Most of the second page also contains problem specification data which is self explanatory. Following the guessed nondimensional stand-off distance (DELTA) and transformed stand-off distance (DELTA) and transformed stand-off distance (DTIL), a listing of the dimensional stand-off distance computed at each iteration is given if the line-continumm radiation model is used.

Species number densities for those species used in the radiation calculation during the final iteration are printed on the third page. The fourth page provides an output of radiative fluxes computed during the final iteration. The continumm contribution and line contribution to the spectral flux is printed for three ETA points (ETA = 0.0 = wall, ETA = stagnation point, ETA = 1.0 = shock) as a function of frequency intervals and frequency centers respectively. The columns of fluxes in watts/cm^2 denoted by Q PLUS and Q MINUS designates fluxes toward the surface and away from the surface respectively.

From the fifth until the thirtieth page, results of the overall iteration, either final or intermediate, are presented. The fifth page begins with one of the following messages: (1) "INTERMEDIATE PRINT AT ITERATION NO. a MCONV = b ECONV = c DCONV = d" where a is the overall iteration number, and b, c, and d are either T (true) or F (false) depending upon if the momentum - global continuity equation, energy-species continuity equations, and the shock standoff distance have converged, respectively. If this message is printed at least one of b, c, or d must be F. (2) "SOLUTION CONVERGED IN e ITERATIONS", where e is the number of overall iterations it took to converge. Following either one of these two messages is a printout of the shock stand-off distance parameters DELTA, DTIL; the convective (QC), radiative (QR), diffusive

(QD) and total heating rate with the respective units printed. To the left of the heating rate data the density ratio across the shock (RB) and the mass injection rate (RVW) are stated. Following the heating rate data is a print of some of the solution profiles as a function of the shock layer coordinates ETA and (Y/D). The solution profiles printed are:

F' = velocity function

RV = $\rho v/\rho_\infty U_\infty$ (nondimensional mass flux per unit area)

T/TD = T/T_δ (nondimensional temperature)

E = 1E (radiative flux divergence)

V = V/U_∞ (nondimensional normal velocity)

V(FT/SEC) = dimensional normal velocity

G = nondimensional total enthalpy

H(STATIC) = nondimensional static enthalpy

These profiles appear in part of page 5 and in page 6.

The shock layer thermodynamic and transport properties as a function of ETA and Y/D are printed in the following pages. The profiles printed are:

P(ATM) = pressure in atmospheres

T(DEG. KEL.) = temperature in degrees Kelvin

RHO (SLUGS/FT3) = density in slugs/cubic feet

M (LBM/FT-SEC) = viscosity in lbm/(ft-sec)

RM (LBF2-SEC3/FT6) = product of density and viscosity in $lbf^2 - sec^3/ft^6$

K(BTU/FT-SEC-R) = thermal conductivity in Btu/(ft-sec-°R)

The next four pages present the profiles of the mass fractions of O_2, N_2, O, N, O^+. N^+, e^-, C, H, H_2, CO, C_3, CN, C_2H, and C_2H_2 as a function of ETA. Page 13 contains the profiles of mixture specific heat at

constant pressure (CP), mass fractions for C_3H, C_4H, HCN and C^+, and the mixture molecular weight (AMW) as a function of ETA.

The above description is for a standard output. However, if the intermediate print option is used (i.e. IDEBUG > 0) or if the thermodynamic fits and intermediate results from the chemical equilibrium calculation are desired (i.e. NDEBUG = 1), additional information is printed.

If message number (2) is printed, the run has ended and the program proceeds to the next case or stops if there are no additional cases to be run. If message number (1) is printed, the solution has not converged and additional iterations must be carried out.

Punched Card Output

After each overall iteration the program outputs a deck of cards containing all relevant data on flight conditions, body characteristics, and shock layer structure. These data are punched by a subroutine named PUNCH in such a manner that the cards may be used to restart a run at a later time. For example, if the program is allowed to run for 3 hours, during that time the program might perform 4 overall iterations and it would output 4 decks of punched cards, one for each overall iteration. If the solution has not converged after the 3 hours have run out, the fourth deck of punched cards may be used to restart the run at a later time.

REFERENCES

A.1. Del Valle, E.G., and R.W. Pike, "Computation of the Equilibrium Composition of Reacting, Gas-Solid Mixtures with Material and Energy Balance Constraints," NASA-RFL-10, March 1970, Reacting Fluids Laboratory, Louisiana State University, Baton Rouge, Louisiana.

A.2 Engel, Carl D., Ablation and Radiation Coupled Viscous Hypersonic Shock Layers, Ph.D. Dissertation, Louisiana State University, Baton Rouge, Louisiana (1971).

A.3 Esch, Donald D., Stagnation Region Heating of a Phenolic-Nylon Ablator During Return from Planetary Missions, Ph.D. Dissertation, Louisiana State University, Baton Rouge, Louisiana (1971).

APPENDIX B

SLAC PROGRAM LISTING

```
C         ****                      S L A C                       ****
C         STAGNATION LINE HEATING ANALYSIS FOR A  VISCOUS HYPERSONIC
C         SHOC LAYER WITH FINITE-RATE OR EQUILIBRIUM CHEMISTRY AND
C         RADIATIVE HEAT TRANSFER.
C                  G. PEREZ, C.D. ENGEL, AND D.D. ESCH        JUNE, 1972
C
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,ECONV,DCONV,LT,IAB
      LOGICAL MCONV,ECONV,DCONV
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMON/NUMBER/NSP,NNS,NE,NC
      COMMCN/EQ2/AA(20,5),ICODE(20)
      COMMON/EQ3/IA(20,5)
C
C
C
C     **           D R I V E R    P R O G R A M               **
C
C
 1    CONTINUE
C
C     **  READ AND PRINT ALL INPUT DATA  **
C
      CALL INPUT
C
C-----FLOAT AA(I,J) MATRIX....
C
      DO30I=1,NSP
      DO30J=1,NE
 30   AA(I,J)=IA(I,J)
C
C     **  COMPUTE NECESSARY INITIAL QUANTITIES  **
C
      IF(IEM.EQ.0)CALL INIT
      IF(ITYPE.EQ.0)CALL RADIN
```

```
C
C    ****     START OVERALL ITERATION     ****
C
 1000 CONTINUE
      IEM = IEM+1
C
C      **  SOLVE MOMENTUM EQUATION  **
C
      CALL MOMTM
C
C
C
C      **  SOLVE ENERGY AND SPECIES EQUATIONS  **
C
      CALL COUPLE
C
C      **  INTERMEDIATE PRINTOUT  **
C
      CALL OUTPUT (2)
      CALL PONCH
C
C
C
C      **  CHECK SIMULTANEOUS MOMENTUM AND ENERGY CONVERGENCE  **
C
      IF(IEM.GT.MAXD) GO TO 3000
      IF(.NOT.MCONV) GO TO 1000
      IF(.NOT.ECONV) GO TO 1000
      IF(.NOT.DCONV) GO TO 1000
C      **  PRINT ALL OUTPUT  **
C
      CALL OUTPUT ( 1 )
C
C      **  CONVERGED . GO BACK TO RUN ANOTHER CASE  **
C
```

```
      GO TO 1
C
 3000 CONTINUE
C
C     **  MOMENTUM AND ENERGY DID NOT CCNVERGE SIMULTANEOUSLY  **
C
      CALL OUTPUT (3)
      GO TO 1
      END
```

```
      SUBROUTINE INPUT
C
C
C     **  ROUTINE TO READ AND PRINT ALL INPUT DATA  **
C
C
      COMMON /CONV/ FPRCT,TPRCT,DDAMP,TDAMP,PDTIL
      COMMON/CONV1/HDAMP
      COMMON /DEL/ DELTA,DTIL,DTILS
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,ECONV,DCONV,LT,IAB
      COMMON /NON/RDZ,MUDZ,RWDZ,AKNF,HNF,CPNF
      COMMON/PROP1/PI(60),RHO(60), T(60),AMW(60),C (20,60),CC(5,60)
      COMMON/PROP2/  MU(60),RM(60), AK(60)
      COMMON/PROP3/CPS(20,60),HS(20,60),CP (60),HM(60)
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMON /VEL/ F(60),FC(60),Z(60),V(60)
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      COMMON/WALL/RVW,PRW,TWOLD,FLUX(20),CWALL(20),ECWALL(5)
      COMMON /YL/ETA(60),YOND(60)
      COMMON/TIT/TITLE(18)
C
C
      COMMON/EQ1/AI(20), BI(20), CI(20), DI(20), EI(20), FI(20), GI(20),
     X         AII(20),BII(20),CII(20),DII(20),EII(20),FII(20),GII(20)
      COMMON/EQ2/AA(20,5),ICODE(20)
      COMMON/EQ3/IA(20,5)
      COMMON/ID/SP(20),EL(5)
      COMMON/WT/SMW(20),AWT(5)
      COMMON/NUMBER/NSP,NNS,NE,NC
      COMMON/SP1/SS,TOL,NDBUG
C
      REAL  MU,MUDZ
      LOGICAL MCONV,ECONV,DCONV
      DATA END /'END '/
```

```
C
C      **   INPUT FORMATS   **
C
 100   FORMAT (18A4,I8)
 101   FORMAT (9I5,2E12.0,2X,I1)
 102   FORMAT ( 6E12.0 )
 107   FORMAT(I5,  5X,E10.4,I5)
 108   FORMAT(5E15.8)
C
C      **   OUTPUT FORMATS   **
C
 200   FORMAT ( 1H1 , 18A4,I8 //// )
 201   FORMAT ( 12H0 INPUT DATA /// )
 202   FORMAT ( 9H0KEEP    = I5
     1         / 9H NETA    = I5
     2         / 9H MAXM    = I5
     3         / 9H MAXE    = I5
     4         / 9H MAXD    = I5
     5         / 9H FPRCT   = 1PE15.6
     6         / 9H TPRCT   = E15.6
     7         / 9H LT      = I5
     8         / 9H IDEBUG  = I5
     9         / 9H IPHI    = I5   )
  203 FORMAT(1X/'  **  FINITE-RATE CHEMISTRY  **'//)
 204   FORMAT ( 9H0UINF    = 1PE15.6
     1         / 9H RINF    = E15.6
     2         / 9H R       = E15.6
     3         / 9H TW      = E15.6
     4         / 9H HTOTAL  = E15.6
     5         / 9H RVW     = E15.6
     6         / 9H PDTIL   = E15.6 //)
 205   FORMAT ( 9H0NDBUG   = I5
     1         / 9H TOL     = F5.3//)
C
 206   FORMAT ( 20H0INITIAL T PROFILE  / ( 1H , 12F10.5 )     )
```

```
 207  FORMAT ( 20H0INITIAL RHO PROFILE/ ( 1H , 12F10.5 )      )
 208  FORMAT ( 20H0INITIAL RM  PROFILE/ ( 1H , 12F10.5 )      )
 209  FORMAT ( 20H0ETA                / ( 1H , 12F10.5 )      )
 210  FORMAT(32H * CONVECTIVE CALCULATION ONLY * )
 211  FORMAT(36H * UNCOUPLED RADIATION CALCULATION * )
 212  FORMAT(34H * COUPLED RADIATION CALCULATION * )
 213  FORMAT(36H * CONTINUUM AND LINE CALCULATION * )
 214  FORMAT(19H * EMISSION MODEL * )
  215 FORMAT(1X/'  **  EQUILIBRIUM CHEMISTRY   **'//)
 216  FORMAT (16H SPECIES INPUTS
     1       /16H NO. ELEMENTS = I5
     2       /25X,5(I5,2X,A4)     )
 218  FORMAT (16H NO. SPECIES   = I5)
 220  FORMAT (25X,5(I5,2X,A4))
 222  FORMAT (16H NO. SOLIDS    = I5)
C
C CARD 1 -----------------------------------------------------------------
      READ (5,100) TITLE,IEM
      IF ( TITLE ( 1 ) .EQ. END ) STOP
      IF(IEM)6999,5999,6999
 5999 CONTINUE
C
C         ****    START FROM SCRATCH    ****
C
C
C     **  INPUT OPTION PARAMETERS  **
C
C     **  IRAD = 1  NO RADIATION CALCULATED
C         IRAD = 2  UNCOUPLED SOLUTION
C         IRAD = 3  COUPLED SOLUTION  **
C     **  ITYPE=0 SPECTRAL MODEL WITH LINES
C         ITYPE=1 EMISSION MODEL
C
      KETA = NETA
C CARD 2 -----------------------------------------------------------------
```

```
      READ (5,101)KEEP,NETA,IRAD,ITYPE,MAXM,MAXE,MAXD,LT,IPHI,
     1              FPRCT,TPRCT,IDEBUG
      META = NETA
      IF(KETA .EQ. 0) KEEP = 0
      IF(KEEP. GT. 0) NETA = KETA
C
      HDAMP = 0.6
      TDAMP = 0.06
      DDAMP = 0.5
      IF(MAXM .EQ. 0) MAXM=15
      IF(MAXE.EQ. 0) MAXE=5
      IF(MAXD.EQ. 0) MAXD=15
      IF(FPRCT.EQ. 0.0) FPRCT=.005
      IF(TPRCT .EQ. 0.0) TPRCT=0.005
      IF ( NETA .EQ. 0 ) NETA = 51
      IF (IRAD.EQ.0) IRAD =1
C
C
C     **  FREE-STREAM FLIGHT CONDITIONS  **
C CARD 3 ------------------------------------------------------------
      READ(5,102) U INF,R INF,R,TWK,HTOTAL,RVW
      UINF2=UINF**2
      IF(KEEP.GT. 0) TWOLD = T(1)
      T(1)=TWK
C
      IF(HTOTAL .EQ. 0.0) HTOTAL=UINF2/2.0
C
C     **  INITIAL SHOCK QUANTITY ESTIMATES  **
C CARD 4 ------------------------------------------------------------
      READ(5,102) DELTA,DTIL,RZB,RE,PDTIL
      IF(PDTIL.EQ.0.0) PDTIL = .001
C
C     **   INPUT INITIAL TEMPERATURE PROFILE   **
C CARD 5 ------------------------------------------------------------
      IF(LT .EQ. 0) GO TO 2800
```

```
      READ ( 5,102 ) ( T ( I ) , I = 1 , NETA )
 2800 CONTINUE
C     ** INPUT RHO AND (RHO)(MU) PROFILES  **
C CARD 6 -----------------------------------------------------------
      IF(LT.LT.2) GO TO 2900
      READ(5,102) (RHO(I),I=1,NETA)
      READ(5,102) (RM (I),I=1,NETA)
 2900 CONTINUE
C
C     **  SHOCK SHAPE (DEPS/DXI)  **
C
      IF (IPHI .NE. 0 ) GO TO 2550
      DEPS = 0.0
      GO TC 2570
 2550 CONTINUE
C CARD 7 -----------------------------------------------------------
      READ(5,102)  DEPS
 2570 CONTINUE
      DPHI = 1. -DEPS
C
C
      IF ( META .GT. 0 ) GO TO 1000
      IF(KEEP .GT. 0) GO TO 1500
C
C     **  FIXED GRID SIZE CN ETA  **
C
      DETA = 0.02
      ETA ( 1 ) = 0.0
      DO 500 I = 2 , 51
      ETA ( I ) = ETA ( I-1 ) + DETA
 500  CONTINUE
C
      GO TO 1500
C
 1000 CONTINUE
```

```
C
C      **  INPUT ETA POINTS  **
C CARD 8 -----------------------------------------------------------------
      READ ( 5,102 ) ( ETA ( I ) , I = 1 , NETA )
C
 1500 CONTINUE
C
C-----READ SPECIES PARAMETER CARDS.......
C CARD 9 -----------------------------------------------------------------
      READ 107, NDBUG,TOL,IAB
C
      IF(TOL.LE.0.0) TOL = .001
C     NDBUG=OPTIONAL OUTPUT VARIABLE
C     NC = NUMBER OF GASEOUS COMPONENTS
C CARD 10 ----------------------------------------------------------------
      READ 108,(CWALL(I),I=1,NSP)
      GO TO 7999
 6999 CONTINUE
C
C         ****    RE-START    ****
C
      READ(5,1801)  FPRCT,TPRCT,DDAMP,TDAMP,PDTIL,HDAMP
      READ(5,1801) DELTA,DTIL,DTILS
      READ(5,1802)     U INF, RINF, UINF2, R , RE, ITM, IEM, NETA
      READ(5,1803) KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,ECONV,DCONV,LT,IAB
      READ(5,1801)RDZ,MUDZ,RMDZ,AKNF,HNF,CFNF
      READ(5,1804)  NSP,NNS,NE,NC
      READ(5,1801) (PI(J),RHO(J),T(J),J=1,NETA)
      READ (5,1801)(AMW(J),J=1,NETA)
      READ(5,1801) ((C(I,J),J=1,NETA),I=1,NSP)
       READ(5,1801)( HM(J),J=1,NETA)
      READ(5,1801)  (E(J),J=1,NETA)
      READ(5,1804) IRAD,ITYPE
      READ(5,1801)DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      READ (5,1801)(F(J),FC(J),Z(J),V(J),J=1,NETA)
```

```
      READ(5,1801)RVW,PRW,TWOLD                                          INPU2110
      READ(5,1801) (CWALL(I),I=1,NSP)                                    INPU2120
      READ(5,1801) (ECWALL(K),K=1,NE)                                    INPU2130
      READ(5,1801) (ETA(J),YOND(J),J=1,NETA)                             INPU2140
       READ(5,715)NDBUG,IAB,TOL                                          INPU2150
      READ (5,1801)((CC(K,J),J=1,NETA),K=1,NE)                           INPU2150
 1801 FORMAT(6E13.5)                                                     INPU2170
 1802 FORMAT(5E12.5,3I3)                                                 INPU2180
 1803 FORMAT(5I5,3L3,2I5)                                                INPU2190
 1804 FORMAT(4I4)                                                        INPU2200
  715  FORMAT(2I5,E15.6)                                                 INPU2210
C                                                                        INPU2220
 7999 CONTINUE                                                           INPU2230
C                                                                        INPU2240
C        ****    PRINT DATA    ****                                      INPU2250
C                                                                        INPU2260
      WRITE ( 6,200 ) TITLE,IEM                                          INPU2270
      WRITE ( 6,201 )                                                    INPU2280
      WRITE (6,202)KEEP,NETA,MAXM,MAXE,MAXD,FPRCT,TPRCT,LT,IDEBUG        INPU2290
     1             ,IPHI                                                 INPU2300
      WRITE(6,204) U INF,R INF,R,TWK,HTOTAL,RVW,PDTIL                    INPU2310
      IF (IRAD.EQ.1) WRITE (6,210)                                       INPU2320
      IF (IRAD.EQ.2) WRITE (6,211)                                       INPU2330
      IF (IRAD.EQ.3) WRITE (6,212)                                       INPU2340
      IF(IRAD.EQ.1) GO TO 300                                            INPU2350
      IF( ITYPE.EQ.0) WRITE(6,213)                                       INPU2360
      IF( ITYPE.EQ.1) WRITE(6,214)                                       INPU2370
 300  CONTINUE                                                           INPU2380
      WRITE ( 6,206 ) ( T ( I ) , I = 1 , NETA )                         INPU2390
      IF(IFM.EQ.0)T(1) = TWK                                             INPU2400
      WRITE(6,207)(RHO(I),I=1,NETA)                                      INPU2410
      WRITE(6,208)(RM (I),I=1,NETA)                                      INPU2420
      WRITE(6,209)(ETA(I),I=1,NETA)                                      INPU2430
      WRITE(6,217)  DEPS                                                 INPU2440
 217  FORMAT(9H0DEPS/DXI  /(1H ,12F10.5) )                               INPU2450
```

```
      IF(IAB.EQ.0)WRITE(6,203)
      IF(IAB.EQ.1)WRITE(6,215)
      WRITE(6,205)NDBUG,TOL
      WRITE(6,216) NE,(I,EL(I),I=1,NE)
      WRITE(6,218) NSP
      JJ = 1
      KK = JJ+4
 30   WRITE(6,220) (I,SP(I),I=JJ,KK)
      IF(KK+5.GT.NSP) GO TO 35
      JJ = JJ+5
      KK = JJ +4
      GO TC 30
 35   KD = NSP -KK
      IF(KC.LE.0) GO TO 45
      KK = KK +KD
      JJ = JJ +5
 40   GO TO 30
 45   CONTINUE
      WRITE(6,222) NNS
      PRINT 305
305   FORMAT(/' SPECIES'/' NAME',9X,'SMW',                       2X,
     1'WALL MASS FRACTION'/)
      DO 10 I=1,NSP
10    PRINT302,SP(I),SMW(I),CWALL(I)
302   FORMAT(1X,A4,1F13.3,E12.4)
      IF(NDBUG.EQ.0)GOTO9999
      PRINT309
309   FORMAT(//,' SPECIES',35X,'THERMO-CONSTANTS A-G',29X,'RANGE')
      DO 11 I=1,NSP
      PRINT303,SP(I),AII(I),BII(I),CII(I),DII(I),EII(I),FII(I),GII(I)
11    PRINT304,      AI (I),BI (I),CI (I),DI (I),EI ( I),FI (I),GI (I)
303   FORMAT(/,1X,A4,7E12.4,'  LOW RANGE')
304   FORMAT(      5X,7E12.4,'  HIGH RANGE')
      PRINT 307
307   FORMAT(//,25X,' AA(I,J) MATRIX',/)
```

```
      DO12J=1,NE                                                        INPU2810
 12   PRINT 306,(IA(I,J),I=1,NSP)                                       INPU2820
 306  FORMAT(5X,20I5)                                                   INPU2830
C                                                                       INPU2840
 9999 CONTINUE                                                          INPU2850
      RETURN                                                            INPU2860
      END                                                               INPU2870
```

```
      SUBROUTINE INIT
C
C
C     **  ROUTINE TO COMPUTE NECESSARY INITIAL QUANTITIES
C
C
      COMMON /CONV/ FPRCT,TPRCT,DDAMP,TDAMP,PDTIL
      COMMON /DEL/ DELTA,DTIL,DTILS
      COMMON/EQ2/AA(20,5),ICODE(20)
      COMMON/EQ3/IA(20,5)
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON/GUESS/TG1(60),TG2(60)
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,ECONV,DCONV,LT,IAB
      COMMON /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CFNF
      COMMON/PROP1/PI(60),RHO(60), T(60),AMW(60),C (20,60),EC(5,60)
      COMMON/NUMBER/NSP,NNS,NE,NC
      COMMON/PROP2/  MU(60),RM(60), AK(60)
      COMMON/PROP3/CPS(20,60),HS(20,60),CP (60),HM(60)
      COMMON/VECTOR/ CA(60),CB(60),CC(60),B(60)
      COMMON /VEL/ F(60),FC(60),Z(60),V(60)
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      COMMON/SP2/BR,S(20),CSHOCK(5)
      COMMON/WALL/RVW,PRW,TWOLD,FLUX(20),CWALL(20),ECWALL(5)
      COMMON/WT/SMW(20),AWT(5)
      COMMON /YL/ETA(60),YOND(60)
      COMMON/DD/D(60)
      COMMON/IT/AC
      REAL  MU,MUDZ
C
      LOGICAL MCONV,ECONV,DCONV
C
      MCONV = .FALSE.
      ECONV = .FALSE.
      DCONV = .FALSE.
```

```
      DO 900 I=1,60
      DO 900 J=1,NSP
  900 C(J,I) = 1.0E-20
C
C
C     **  DETERMINE DENSITY RATIO , REYNOLDS NUMBER
C         FROM INPUTS OR RANKINE HUGONIOT EQS.  **
C
C GUESSED VALUES
      TD = 12000. + .5E-5*(HTOTAL -6.5E+8)
      RZB=.06
C
      T(NETA) = 1.0
      HNF = 2.*778.28*32.172/UINF2
  998 CONTINUE
      PD = (1. -RZB)*RINF *UINF2/2116.
      HD = HTOTAL/(778.28*32.172)
      CPNF = 1.8*778.28*32.172*TD *2.   /UINF2
      AKNF = 1.8*778.28*TD       *RZB/(R*RINF*UINF*UINF2)
      PI(NETA) = PD
      CALL GAS(NETA)
      RZB1=RINF/(RDZ*RHO(NETA) )
      TEST =ABS((RZB-RZB1)/RZB)
      IF(TEST .LT. 0.005) GO TO 999
      RZB=.5*(RZB+RZB1)
      GO TO 998
  999 CONTINUE
      RE = RDZ*UINF*R*32.174 / MUDZ
C
C
C     **  GUESS AT DELTA TO START  **
C
      IF(DELTA .EQ. 0.0) DELTA=0.78*RZB
      IF(DTIL .EQ. 0.0) DTIL=1.1*DELTA  +1.2*RVW
      WRITE(6,200) RZB,RE
```

```
 200  FORMAT(14H0DENSITY RATIO ,5X,12HREYNOLDS NO.  /2E15.6)
      WRITE(6,201) DELTA,DTIL
 201  FORMAT(6H0DELTA,13X,4HDTIL   /2E15.6)
C
 997  CONTINUE
      DO 995 I=1,NETA
      PI(I) = PD
      E(I) =0.0
 995  CONTINUE
C      **  RANKIN-HUGONIOT RELATIONS   **
C
      VD = -RZB
      TW = T(1)
      T(1) = T(1)/TD
C
C      **  STAGNATION POINT LIMIT QUANTITIES  **
C
      DUD = DPHI + RZB*(1.-DPHI)
C NONDIMENSIONALIZING FACTORS
      AKNF = 1.8*778.28*TD       *RZB/(R*RINF*UINF*UINF2)
      CPNF = 1.8*778.28*32.172*TD *2.   /UINF2
C
C GUESSED F AND Z PROFILES
C
      IF(KEEP. GT. 0) GO TO 9
      N = NETA-2
      FD = RZB/(2.*DUD*DTIL)
      FW = -RVW*FD
      F(1) = FW
      DO 2 K=2,NETA
      F(K) = (FD-FW)*ETA(K) + FW
   2  CONTINUE
      DO 3 I=1,N
      Z(I) = ETA(I+1)/DTIL
   3  CONTINUE
```

```
C GUESSED T PROFILES
      IF(KEEP.GT.0)GOTO9
      IF(LT.GT.0) GO TO 11
      IF(RVW.GT.0.0)GOTO7
C NO BLOWING T PROFILE
      TWG1 = .1033
      DO6K = 2,NETA
      TP = TG1(K) +(T(1)- TWG1)
      T(K) = TP -(T(1)- TWG1) * ETA(K)
    6 CONTINUE
      GO TO 11
    7 CONTINUE
      TWG2 = .3325
C BLOWING T PROFILE
      DO8K = 2,NETA
      TP = TG2(K) +(T(1)- TWG2)
      T(K) = TP -(T(1)- TWG2) * ETA(K)
    8 CONTINUE
      GO TO 11
    9 CONTINUE
      DO 10 K=2,NETA
      TP = T(K) +T(1) -TWOLD
      T(K) = TP -(T(1)-TWOLD)*ETA(K)
      WRITE(6,100) T(K),ETA(K)
 10   CONTINUE
 11   CONTINUE
C
C  **  INITIALIZE SHOCK LAYER PARAMETERS FOR VARIABLE STEP SIZE
      DO 810 I=NETA,60
      ETA(I)=1.0
      T(I) = 1.0
      E(I) = 0.0
      PI(I) = PD
      MU(I)=1.0
      CP(I) = CP(NETA)
```

```
      AK(I) = AK(NETA)
      V(I) = VD
      F(I) = FD
      FC(I)=FD
      DO 810 J=1,NSP
      C(J,I) = C(J,NETA)
      HS(J,I)=1.0
  810 CONTINUE
 1000 CONTINUE
      DO223J=1,60
      DO223K=1,NE
  223 EC(K,J) = 1.E-20
C
      DO221I=1,NSP
      C(I,1) = CWALL(I)
  221 CONTINUE
C-----CALCULATE AMW(N)
C
      WAMW = 0.0
      DO 25 J=1,NSP
  25. WAMW = WAMW +CWALL(J)/SMW(J)
      WAMW = 1./WAMW
  26  AMW(1)= WAMW
C
C
C-----CONVERT WALL AND SHOCK COMPOSITIONS TO AN ELEMENTAL BASIS.....
C
      DO331J=1,NE
      EC(J,1)=0.0
      EC(J,NETA)=0.0
      DO33I=1,NSP
      FAC=AA(I,J)*AWT(J)/SMW(I)
      FC(J,1)=EC(J,1) + FAC*C(I,1)
 33   EC(J,NETA)=EC(J,NETA) + FAC*C(I,NETA)
 331  ECWALL(J)=EC(J,1)
```

```
C
      DO34N=NETA,60
      DO34J=1,NE
 34   EC(J,N)=EC(J,NETA)
C
C        ****    COMPUTE MUDZ AND RMDZ    ****
      AC = 8.128E-08*(TD**1.659)/(PI(1)*R*UINF)
      DO1623J=1,NETA
 1623 D(J) = AC*(T(J)**1.659)
      CALL ELRAT
      CALL CHEMEQ (1,NETA)
      MUDZ = 1.0
      CALL PROPRT (NSP,NETA,NETA)
      MUDZ = MU(NETA)
      RMDZ = RM(NETA)
      CALL PROPRT (NSP,1,NETA)
      DTILS = .01
      IF(IDEBUG .EQ. 0) RETURN
      WRITE(6,4000) VD,DUD,PD
      WRITE(6,4000) DELTA,DTIL,RZB,RE
 4000 FORMAT(1H0,6E15.6)
 203  FORMAT(6E12.0)
  100 FORMAT(1X,9E14.6)
      RETURN
      END
```

```
      SUBROUTINE MOMTM
C
C -----THIS SUBROUTINE SOLVES THE MOMENTUM EQUATION AS A
C       SECOND ORDER EQUATION AND A FIRST ORDER EQUATION -----
      COMMON /CONV/ FPRCT,TPRCT,DDAMP,TDAMP,PDTIL
      COMMON /DEL/ DELTA,DTIL,DTILS
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,ECONV,DCONV,LT,IAB
      COMMON /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CFNF
      COMMON/PROP1/PI(60),RHO(60), T(60),AMW(60),C (20,60),EC(5,60)
      COMMON/PROP2/  MU(60),RM(60), AK(60)
      COMMON/PROP3/CPS(20,60),HS(20,60),CP (60),HM(60)
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTCTAL
      COMMON/VECTOR/ CA(60),CB(60),CC(60),B(60)
      COMMON /VEL/ F(60),FC(60),Z(60),V(60)
      COMMON/WALL/RVW,PRW,TWOLD,FLUX(20),CWALL(20),ECWALL(5)
      COMMON /YL/ETA(60),YOND(60)
      LOGICAL MCONV,ECONV,DCONV
C
C----- INITIALIZED QUANTITIES -----
C
      MCONV = .FALSE.
      DTILS = DTIL
      ITM = 1
      N = NETA -2
      L = NETA-1
      AA3 = RZB*(1.-RZB)*DPHI**2/DUD
      DTILS2 = DTILS
      IF(IEM.GT.3) DTIL=.5*(DTIL+DTILS2)
C
C
C-----Z''+A1*Z'+A2*Z=A3
C     COMPUTE A1,A2,A3
   14 CONTINUE
```

```
C----- BOUNDARY CONDITICNS -----
C
      RED = RE*DTIL
      RED2 = 2.*RED*DTIL*DUD
      DTIL2 = DTIL*DTIL
      FD = RZB/(2.*DUD*DTIL)
      FW = -RVW*FD
      F(1) = FW
      B(L) = 1./DTIL
      ITER = 1
   15 CONTINUE
      II = 1
      DO 20 I=1,N
      DET=ETA(I+1)-ETA(I)
      DETN=ETA(I+2)-ETA(I+1)
      D1   =  DETN*(DETN+DET)
      D2   =  DETN*DET
      D3   =  DET*(DETN+DET)
      RMP = DET*RM(I+2)/D1 +(DETN-DET)*RM(I+1)/D2 -DETN*RM(I)/D3
      A1 = (RED2*F(I+1) +RMP)/RM(I+1)
      A2 = -RED2*DTIL*Z(I)/RM(I+1)
      A3 = -2.*RED*( AA3/(RHO(I+1)*RM(I+1))
     1                 +DTIL2  *DUD*Z(I)**2/(2.*RM(I+1)) )
C
C-----CA*Z(N-1)+CB*Z(N)+CC*Z(N+1)=B
C     COMPUTE CA,CB,CC
      CA(II) =(2.-A1*DETN)/D3
      CB(I)=A1*(DETN-DET)/D2-2./D2+A2
      CC(I)=(2.+A1*DET)/D1
      B(I)=A3
      II = I
   20 CONTINUE
      B(N)=B(N)-CC(N)/DTIL
C
      CALL TRID (N)
```

```
C
C-----INTEGRATE FIRST ORDER EQUATION-----
      FC(1)=FW
      SUM=FW+ (B(1)+FW)*(ETA(2)-ETA(1))*DTIL/2.
      FC(2)=SUM
      DO 30 K=3,NETA
      SUM=SUM+DTIL*(B(K-1)+B(K-2))*(ETA(K) -ETA(K-1))/2.
   30 FC(K) = SUM
C
C-----CHECK FOR CONVERGENCE
C
      DO 40 K=2,NETA
      PRCT=ABS((FC(K)-F(K))/F(K))
      IF (PRCT.GT.FPRCT) GO TO 50
   40 CONTINUE
      GO TO 90
   50 CONTINUE
      ITER=ITER+1
      DO 60 K=1,NETA
   60 F(K)=FC(K)
      DO 65I=1,N
   65 Z(I)=B(I)
      IF(ITER.GE.MAXM ) GO TO 90
      GO TO 15
   90 CONTINUE
C
C----- COMPUTE NEW DTIL -----
C
      DTILC = (FD-FW)*DTIL/(F(NETA)-FW)
      PRCT = ABS((DTIL-DTILC)/DTIL)
      IF(ITM.GT.MAXM) GO TO 160
      ITM = ITM +1
      IF(PRCT.LE.PDTIL) GO TO 150
      DTIL = DTIL +DDAMP*(DTILC-DTIL)
      GO TO 14
```

```
 150   CONTINUE                                                          MOMT1060
       DTIL = DTIL+ DDAMP*(DTILC -DTIL)                                  MOMT1070
       MCONV = .TRUE.                                                    MOMT1080
C                                                                        MOMT1090
C CHECK MOMENTUM-ENERGY CONVERGENCE                                      MOMT1100
       PRCT = ABS((DTIL-DTILS)/DTILS)                                    MOMT1110
       IF(PRCT.LE.PDTIL) DCCNV = .TRUE.                                  MOMT1120
 160   CONTINUE                                                          MOMT1130
C                                                                        MOMT1140
       DO 170 K=1,NETA                                                   MOMT1150
 170   V(K) = -FC(K)*DTIL*2./RHO(K)                                      MOMT1160
C DEBUG OUTPUT                                                           MOMT1170
       IF(IDEBUG.EQ. 0) RETURN                                           MOMT1180
       WRITE(6,102) ITER,ITM                                             MOMT1190
 102   FORMAT(10X,2I3/)                                                  MOMT1200
       WRITE(6,100) DTIL,DTILC                                           MOMT1210
       WRITE(6,101)                                                      MOMT1220
  101 FORMAT(6X,'ETA',12X,'F',12X,'FC',12X,'RHO',12X,'RM',12X,'VS',12X,  MOMT1230
     1 'V')                                                              MOMT1240
      DO 120 K=1,NETA                                                    MOMT1250
      VS=-FC(K)*DTIL*UINF*2./RHO(K)                                      MOMT1260
      WRITE(6,100) ETA(K),F(K),FC(K),  RHO(K),RM(K),VS ,V(K)             MOMT1270
  100 FORMAT(1X,9E14.6)                                                  MOMT1280
  120 CONTINUE                                                           MOMT1290
      WRITE(6,103)                                                       MOMT1300
  103 FORMAT(6X,'ETA',13X,'Z',13X,'B',12X,2HF')                          MOMT1310
      DO 121 I=1,N                                                       MOMT1320
      U=B(I)*DTIL                                                        MOMT1330
      WRITE(6,100) ETA(I+1),Z(I),B(I),U                                  MOMT1340
  121 CONTINUE                                                           MOMT1350
      RETURN                                                             MOMT1360
      END                                                                MOMT1370
```

```
      SUBROUTINE OUTPUT(N)
C
C
C     **  ROUTINE TO PRINT SHOCK LAYER SOLUTION  **
C
C
      COMMON/ID/SP(20),EL(5)
      COMMON /CONV/ FPRCT,TPRCT,DDAMP,TDAMP,PDTIL
      COMMON /DEL/ DELTA,DTIL,DTILS
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,ECONV,DCONV,LT,IAB
      LOGICAL MCONV,ECONV,DCONV
      COMMON /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CPNF
      COMMON/NUMBER/NSP,NNS,NE,NC
      COMMON/PROP1/PI(60),RHO(60), T(60),AMW(60),C (20,60),EC(5,60)
      COMMON/PROP2/  MU(60),RM(60), AK(60)
      COMMON/PROP3/CPS(20,60),HS(20,60),CP (60),HM(60)
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      COMMON /SFLUX/QRI(3)
      COMMON/VECTOR/ CA(60),CB(60),CC(60),B(60)
      COMMON /VEL/ F(60),FC(60),Z(60),V(60)
      COMMON/WALL/RVW,PRW,TWOLD,FLUX(20),CWALL(20),ECWALL(5)
      COMMON /YL/ETA(60),YOND(60)
      COMMON /DD/ D(60)
      COMMON/SP1/SS,TOL,NDBUG
      DIMENSION BOUT(60),DQR(30)
      REAL   MU,MUDZ
      DATA HEAD1/'WALL'/ ,HEAD2/'    '/,HEAD3/'SHOC'/
C
C     **  COMPUTE RADIATION FLUX IF UNCOUPLED PROBLEM  **
C
      IF(ITYPE.NE.0)GOTO20
      IF(IRAD .EQ. 2  .AND.  N .NE. 2) CALL TRANS(1)
      IF (IRAD.NE.1) CALL TRANS2
```

```
 20    IF(IRAD.EQ.2.AND.ITYPE.EQ.1) CALL EFLUX
       WRITE(6,203)
  203  FORMAT(1H1)
C
C      **  COMPUTE Y COORDINATE  **
C
       YOND(1) = 0.0
       SUM = 0.0
       DO 40 K=2,NETA
       DETA= ETA(K)-ETA(K-1)
       SUM= SUM +DETA*(1./RHO(K)+1./RHO(K-1))/2.
       YOND(K) = DTIL*SUM
  40   CONTINUE
       DELTA = YOND(NETA)
       DO 50 K=1,NETA
       YOND(K)= YOND(K)/DELTA
  50   CONTINUE
C
C      **  COMPUTE CONVECTIVE HEATING RATE  **
C
C WATTS/CM**2
       QC = -AK(1)*RINF*UINF*UINF2*   (T(2)-T(1))/
     1  (.88*778.28        *YOND(2)*DELTA*RZB)
C BTU/FT**2-SEC
       QCP=QC*.88
C
C      **  COMPUTE RADIATIVE FLUX TO SURFACE  **
C
       QR = 0.0
       IF(IRAD .EQ. 1) GO TO 445
       DO 1100 K=2,NETA
       QR = QR + QUAD(YOND,E,K)
 1100  CONTINUE
C WATTS/CM**2
       QR =-QR *RINF*UINF2*UINF *DELTA/(685.*RZB)
```

```
      IF(ITYPE.EQ.0)QR=-QRI(1)
 445  CONTINUE
C BTU/FT**2-SEC
      QRP=QR*0.88
C     **  COMPUTE DIFFUSIVE FLOW TO SURFACE  **
C WATTS/CM**2
      QD = 0.
      DO1768I=1,NSP
 1768 QD = QD + HS(I,1)*(C(I,2) - C(I,1))
      QD = - RHO(1)*D(1)*QD/(2.*DELTA*YOND(2))
      QD = RINF*UINF*UINF2*QD/(.88*778.28)
C BTU/FT**2-SEC
      QDP = .88*QD
      QTOTAL=QC+QR+QD
      QTOTP=QTOTAL*.88
C
C     **  DIMENSIONALIZE RHO,MU,P,AND E  **
C
      DO 450 I = 1 , NETA
      RHO(I)=RHO(I)*RDZ
      MU (I)=MU(I)*MUDZ
      RM (I)=RM(I)*RMDZ
      AK(I) = AK(I)/AKNF
      E(I) = E(I) * RINF    * UINF2 * UINF / (20866.0 * R *RZB)
      CP(I) = CP(I)/CPNF
 450  CONTINUE
C
      GO TO  (1,2,3,4) , N
C
 1    WRITE(6,201) IEM
 201  FORMAT(23H SOLUTION CONVERGED IN  ,I3,11H ITERATIONS //)
      GO TO 4
C
 2    WRITE (6,202) IEM,MCONV,ECONV,DCONV
 202 'FORMAT(1H0,37H INTERMEDIATE PRINT AT ITERATION NO. ,I4,10X,
```

```
     1'MCONV=',L4,5X,'ECONV=',L4,5X,'DCONV=',L4,//)
      GO TO 4
C
    3 CONTINUE
C
 4    CONTINUE
C
C     **  PRINT SHOCK QUANTITIES AND HEATING RATE  **
C
      WRITE(6,204)     DELTA,DTIL
  204 FORMAT(1H0,                  25X,9H DELTA = ,1PE14.6,10X,7HDTIL = ,
     1    E15.6)
C
      WRITE (6,210)        QC,QCP
 210  FORMAT ( 1H0,                25X,5H QC = E15.6,2X,13H(WATTS/CM**2),
     1           2X,1H=,E15.6,2X,17H(BTU/FT**2 - SEC)         )
      WRITE(6,212) RZB,QR,QRP
 212  FORMAT ( 1H0,6H RB  =,F9.4,10X,5H QR = E15.6,2X,13H(WATTS/CM**2),
     1           2X,1H=,E15.6,2X,17H(BTU/FT**2 - SEC)        )
      WRITE(6,213) RVW,QD,QDP
 213  FORMAT ( 1H0,6H RVW =,F9.4,10X,5H QD = E15.6,2X,13H(WATTS/CM**2),
     1           2X,1H=,E15.6,2X,17H(BTU/FT**2 - SEC)        )
C
      WRITE(6,215) QTOTAL,QTOTP
 215  FORMAT(1H0,16HTOTAL HEATING  = E15.6,2X,13H(WATTS/CM**2),
     1        2X,1H=,E15.6,2X,17H(BTU/FT**2 - SEC)      )
C
C     **  PRINT Y/D , F AND T PROFILES  **
C
      WRITE(6,205)
 205  FORMAT(1H0,7X, 4H ETA, 5X, 4HY/DZ,  8X, 2HF', 8X, 3H RV, 8X ,
     1        4HT/TD,  4X ,13H E(WATTS/CM3),4X,2H V, 7X,12H V (FT/SEC) ,
     2  5X,2H G,6X,12H H (STATIC) , //)
      FP = 0.0
      NS=NSP
```

```
C
      DO 100 I=1,NETA
C COMPUTE ENTHALPIES
      HSTAT = 0.0
      DO 99 J=1,NS
   99 HSTAT = HSTAT + HS(J,I)*C(J,I)
      G = HSTAT + V(I)**2
C
      HEAD=HEAD2
      IF(I .EQ. 1) HEAD=HEAD1
      IF (I .EQ. NETA) HEAD=HEAD3
      YDZ = YOND(I)
      IF(I.EQ.NETA) FP =1.0
      RV = -FC(I)*DTIL*2.
      VS = V(I) *UINF
      WRITE(6,208) HEAD,ETA(I),YDZ,FP,RV,T(I),E(I),V(I),VS,G,HSTAT
  208 FORMAT(1H ,A4,  F6.3,1P10E12.3)
      IF(I.LT.NETA-1) FP = Z(I)*DTIL
  100 CONTINUE
C
C     **  WRITE OUT SHOCK LAYER GAS PROPERTIES   **
C
      WRITE(6,44)
   44 FORMAT(1H1,48X,28H-SHOCK LAYER GAS PROPERTIES-    )
C
      WRITE(6,206)
  206 FORMAT(1H0,3X,3HETA,8X,4H Y/D,12X,2HP ,12X,2H T,11X,3HRHO,11X,2HMU
     1       , 12X, 3HRMU,11X,2H K)
C
      WRITE(6,207)
  207 FORMAT (1H ,27X,6H(ATM.),6X,13H (DEG.KEL.)  ,12H(SLUGS/FT3) ,2X,
     1       28H(LBM/FT-SEC) (LBF2-SEC3/FT6) ,16H (BTU/FT-SEC-R) ,//)
C
      DO 101 I=1,NETA
C
```

C-3

```
      TS = T(I)*TD
      WRITE(6,8)ETA(I),YOND(I),PI(I),TS  ,RHO(I),MU(I),RM (I),AK(I)
C
    8 FORMAT(1H F7.4,1P8E14.4)
    9 FORMAT(1H F7.4,1P7E14.4)
C
  101 CONTINUE
C
C     **  WRITE SPECIES MASS FRACTIONS  **
C
      WRITE(6,230)
 230  FORMAT (1H1,48X,26H-SPECIES MASS FRACTIONS-    )
      WRITE(6,231)
 231  FORMAT (1H ,14X,3H O2,11X,2HN2,11X,3H O ,11X,3H N ,11X,3H O+,
     1      11X,3H N+,11X,3H E-,//)
      DO 102 I=1,NETA
      WRITE(6,8) ETA(I),C(1,I),C(2,I),C(3,I),C(4,I),C(5,I),
     1      C(6,I),C(7,I)
 102  CONTINUE
      WRITE(6,230)
      WRITE(6,233) (SP(I),I=8,15)
 233  FORMAT(2X,4H ETA,1X,8(10X,A4)//)
      WRITE(6,8) (ETA(I),      (C(J,I),J= 8,15),I=1,NETA)
      WRITE(6,230)
      WRITE(6,234) (SP(I),I=16,20)
 234  FORMAT(2X,4H ETA,7X,3H CP,5(11X,A4),8X,4H AMW,//)
      WRITE(6,9) (ETA(I),CP(I),(C(J,I),J=16,20),AMW(I),I=1,NETA)
C NONDIMENSIONALIZE
      DO1001I=1,NETA
      RHO(I) = RHO(I)/RDZ
      MU(I) = MU(I)/MUDZ
      RM(I) = RM(I)/RMDZ
      E(I)=((E(I)*R)/(RINF*UINF**3))*20866.0*RZB
      CP(I)=CP(I)*CPNF
 1001 AK(I)=AK(I)*AKNF
```

```
  217  FORMAT(6E12.5)
 1000 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE PONCH
      COMMON /CONV/ FPRCT,TPRCT,DDAMP,TDAMP,PDTIL
      COMMON/CONV1/HDAMP
      COMMON /DEL/ DELTA,DTIL,DTILS
      COMMCN /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,E CONV,DCONV,LT,IAB
      COMMON /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CPNF
      COMMON/NUMBER/NSP,NNS,NE,NC
      COMMON/PROP1/PI(60),RHO(60), T(60),AMW(60),C (20,60),EC(5,60)
      COMMON/PROP2/  MU(60),RM(60), AK(60)
      COMMON/PROP3/CPS(20,60),HS(20,60),CP (60),HM(60)
      COMMON /RFLUX/ E(6C),IRAD,ITYPE
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTCTAL
      COMMON /VEL/ F(60),FC(60),Z(60),V(60)
      COMMCN/WALL/RVW,PRW,TWOLD,FLUX(20),CWALL(20),ECWALL(5)
      COMMON /YL/ETA(60),YOND(60)
      COMMON/TIT/TITLE(18)
      COMMON/SP1/SS,TOL,NDBUG
      REAL  MU,MUDZ
C
C        ****    PUNCH-OLT CARDS FOR RESTART     ****
C
      WRITE(7,566)TITLE,IEM
  566 FORMAT(18A4,I8)
      WRITE(7,1801) FPRCT,TPRCT,DDAMP,TDAMP,PDTIL,HDAMP
      WRITE(7,1801)DELTA,DTIL,DTILS
      WRITE(7,1802)   U INF, RINF, UINF2, R , RE, ITM, IEM, NETA
      WRITE(7,1803)KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,E CONV,DCONV,LT,IAB
      PUNCH 1801, RDZ,MUDZ,RMDZ,AKNF,HNF,CPNF
      WRITE(7,1804) NSP,NNS,NE,NC
      WRITE(7,1801)(PI(J),RHO(J),T(J),J=1,NETA)
      WRITE(7,1801)(AMW(J),J=1,NETA)
      WRITE(7,1801)((C(I,J),J=1,NETA),I=1,NSP)
      WRITE(7,1801)( HM(J),J=1,NETA)
      WRITE(7,1801) (E(J),J=1,NETA)
```

```
      WRITE(7,1804)IRAD,ITYPE
      PUNCH 1801, DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      WRITE(7,1801)(F(J),FC(J),Z(J),V(J),J=1,NETA)
      PUNCH 1801, RVW,PRW,TWOLD
      WRITE(7,1801)(CWALL(I),I=1,NSP)
      WRITE(7,1801)(ECWALL(K),K=1,NE)
      WRITE(7,1801)(ETA(J),YCND(J),J=1,NETA)
      WRITE(7,715)NDBUG,IAB,TOL
      WRITE(7,1801)((EC(K,J),J=1,NETA),K=1,NE)
 1801 FORMAT(6E13.5)
 1802 FORMAT(5E12.5,3I3)
 1803 FORMAT(5I5,3L3,2I5)
 1804 FORMAT(4I4)
  715  FORMAT(2I5,E15.6)
      RETURN
      END
```

```
      SUBROUTINE GAS (KODE )
C
C     **   THERMODYNAMIC AND TRANSPORT PROPERTIES OF AIR **
C     **  REFERENCE NASA TR R-50 **
C
C  THE FOLLOWING PROPERTIES ARE CALCULATED
C                 TEMPERATURE AT WHICH PROPERTIES ARE WANTED (T) IN DEG R
C                 PRESSURE AT WHICH PROPERTIES ARE WANTED (P) IN LB/IN**2
C                 RATIO OF SPECIFIC HEATS (GAMMA) IN DIMENSIONLESS
C                 SPECIFIC HEAT AT CONSTANT PRESSURE (CP) IN BTU/LB-DEG R
C                 ABSOLUTE VISCOSITY (V) IN LB/FT-SEC
C                 PRANDTL NUMBER (PR) IN DIMENSIONLESS
C                 THERMAL CONDUCTIVITY (XK) IN BTU/FT-SEC-DEG R
C                 PRESSURE (P) IN ATMOSPHERES
C                 DENSITY (DEN) IN LB/FT**3
C                  ENTHALPY (H) IN BTU/LB
C                  ENTROPY (S) IN BTU/LB-DEG R
C                  COMPRESSIBILITY (Z) IN DIMENSIONLESS
C                  SPEED OF SOUND (SOS) IN FT/SEC
C                  SPECIFIC HEAT AT CONSTANT VOLUMN (CV) IN BTU/LB-DEG R
C                  ENTROPY (S) IN FT**2/SEC**2
C                  VELOCITY (VEL) IN FT/SEC
C                  PRESSURE (P) IN LBS/FT**2
C                  MACH NUMBER (M) IN DIMENSIONLESS
C
C
C NOMENCLATURE  1=OXYGEN MOLECULES, 2=NITROGEN MOLECULES, 3=OXYGEN ATMOS
C               4=NITROGEN ATMOS, 5=OXYGEN IONS, 6=NITROGEN IONS
C               7=ELECTRONS
C
C
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CFNF
      COMMON/PROP1/PI(60),RHO(60),TI(60),AMW(60),C (20,60),CC(5,60)
      COMMON/PROP2/  MU(60),RM(60), AK(60)
```

```
      COMMON/PROP3/CPS(20,60),HS(20,60),CPT(60),HM(60)
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      COMMON/WALL/RVW,PRW,TWOLD,FLUX(20),CWALL(20),ECWALL(5)
      REAL  MU,MUDZ
      LOGICAL MCONV,GCONV,SCONV
      DATA GASC /49721.7/
C
C
      DO 2000 I=KODE,NETA
      T = TI(I) * TD
      P = PI(I)
C
C THE FOLLOWING PART OF PROGRAM USES PRESSURE IN ATMOSPHERES
C AND TEMPERATURE IN DEG K
C
      ITER=0
C
C     **  TEMPERATURE - ENTHALPY ITERATION  **
C
 900  CONTINUE
      ITER=ITER+1
      IF(T.LT.100.) T=100.
      A1=11390./T
      A2=18990./T
      A3=2270./T
      A4=3390./T
      A5=228./T
      A6=326./T
      A7=22800./T
      A8=48600./T
      A9=27700./T
      A10=41500./T
      A11=38600./T
      A12=58200./T
      A13=70.6/T
```

```
      A14=188.9/T
      A15=22000./T
      A16=47000./T
      A17=67900./T
      A18=2270./(4.*T)
      A19=TANH(A18)
      A20=3390./(4.*T)
      A21=TANH(A20)
      TT=1./T
      TSQ=T**2
      TSQRT=T**.5
      A22=112.2222/T
      A23=T/59000.
      A24=T/113200.
      A25=T/75400.
      AA1=EXP(-A1)
      AA2=EXP(-A2)
      AA3=EXP(A3)
      AA4=EXP(A4)
      AA5=EXP(-A5)
      AA6=EXP(-A6)
      AA7=EXP(-A7)
      AA8=EXP(-A8)
      AA9=EXP(-A9)
      AA10=EXP(-A10)
      AA11=EXP(-A11)
      AA12=EXP(-A12)
      AA13=EXP(-A13)
      AA14=EXP(-A14)
      AA15=EXP(-A15)
      AA16=EXP(-A16)
      AA17=EXP(-A17)
C  CALCULATING ENERGIES PER COMPONENT OF GAS MIXTURE ABOVE
C  REFERENCE ENERGIES.
       E1=2.5+((2.*AA1*A1+AA2*A2)/(3.+2.*AA1+AA2))+(A3/(AA3-1.))
```

```
      ET=2.5+ (A4/(AA4-1.))
      E3=1.5+((3.*AA5*A5+AA6*A6+5.*AA7*A7+AA8*A8)/(5.+3.*AA5+AA6+5.*AA7+
     1AA8))
      E4=1.5+((10.* AA9*A9+6.*AA10*A10)/(4.+10.*AA9+6.*AA10))
      E5=1.5+((10.*AA11*A11+6.*AA12*A12)/(4.+10.*AA11+6.*AA12))
      E6=1.5+((3.*AA13*A13+5.*AA14*A14+5.*AA15*A15+AA16*A16+5.*AA17*A17)
     1/(1.+3.*AA13+5.*AA14+5.*AA15+AA16+5.*AA17))
      E7=1.5
C TOTAL ENERGY PER COMPONENT OF GAS MIXTURE
      EN1=E1
      EN2=ET
      EN3=E3+29500./T
      EN4=E4+56600./T
      EN5=E5+187500./T
      EN6=E6+225400./T
      EN7=E7
C LOGS OF PARTITION FUNCTIONS
      TL1=ALOG(T)*3.5
      TL2=ALOG(T)*2.5
      EQ1=TL1+.11+ALOG((3.+2.*AA1+AA2)/(1.-(1.0/AA3)))
      EQ2=TL1-.42-ALOG((1.-(1.0/AA4)))
      EQ3=TL2+.5+ALOG((5.+3.*AA5+AA6+5.*AA7+AA8))
       EQ4=TL2+.3+ALOG((4.+10.*AA9+6.*AA10))
      EQ5=TL2+.5+ALOG((4.+10.*AA11+6.*AA12))
       EQ6=TL2+.3+ALOG((1.+3.*AA13+5.*AA14+5.*AA15+AA16+5.*AA17))
      EQ7=TL2-14.24
C EQUILIBRIUM CONSTANS FOR CHEMICAL REACTIONS
      EK1=-59000./T+2.*EQ3-EQ1
      EK2=-113200./T+2.*EQ4-EQ2
      EK3=-158000./T+EQ5+EQ7-EQ3
      EK4=-168800./T+EQ6+EQ7-EQ4
      CCC=-79.9
      IF(EK1.LE.CCC) EK1=-79.9
      IF(EK2.LE.CCC) EK2=-79.9
      IF(EK3.LE.CCC) EK3=-79.9
```

```
      IF(EK4.LE.CCC) EK4=-79.9
      XK1=EXP(EK1)
      XK2=EXP(EK2)
      XK3=EXP(EK3)
      XK4=EXP(EK4)
      XK34=.2*XK3+.8*XK4
      EE1=(-0.8+(.64+.8*(1.+((4.*P)/XK1)))**0.5)/(2.*(1.+4.*P/XK1))
      EE2=(-0.4+(.16+3.84*(1.+(4.*P)/(XK2)))**0.5)/(2.*(1.+4.*P/XK2))
      EE3= 1./(( 1.+P/XK34)**.5)
      IF(EE1.GT..19999)EE1 = .19999
      IF(EE2.GT..79999)EE2 = .79999
      IF(EE3.GT..99999)EE3 = .99999
C COMPRESSIBILITY (Z) DIMENSIONLESS
      Z=1.+EE1+EE2+2.*EE3
C COMPONENT MOL FRACTIONS IN AIR
      X1=(.2-EE1)/Z
      X2=(.8-EE2)/Z
      X3=(2.*EE1-.4*EE3)/Z
      X4=(2.*EE2-1.6*EE3)/Z
      X5= .4*EE3/Z
      X6= 1.6*EE3/Z
      X7= 2.*EE3/Z
      IF(X1.LF.0.)  X1=1.E-20
      IF(X2.LE.0.)  X2=1.E-20
      IF(X3.LE.0.)  X3=1.E-20
      IF(X4.LE.0.)  X4=1.E-20
      IF(X5.LE.0.)  X5=1.E-20
      IF(X6.LE.0.)  X6=1.E-20
      IF(X7.LE.0.)  X7=1.E-20
C ENERGY PER MOL OF INITIALLY UNDISSOCIATED AIR-DIMENSIONLESS
      FR= Z*(X1*EN1+X2*EN2+X3*EN3+X4*EN4+X5*EN5+X6*EN6+X7*EN7)
C ENTHALPY PER INITIAL MOL OF AIR-DIMENSIONLESS
      HR=ER+Z
C ENTHALPY PER INITIAL MOL OF AIR (H) IN BTU/LB
      H=HR*T*.12348
```

```
      IF(KODE.LT.NETA) GO TO 1000
      HRATO=.5*(H-HD)/H
      AHR  = ABS( HRATO )
      IF(AHR .LE. 0.0010) GO TO 999
      IF(ITER .GT.1) GO TO 203
      TP=T
      HP=HRATO
      T = T *(1. - HRATO )
      IF(ITER .LT. 15) GO TO 900
  203 CONTINUE
      TS=T*(1.0-HRATO)
      IF(HRATO*HP .LT.0.0) TS=.5*(T+TP)
      TP=T
      T=TS
      HP=HRATO
      IF(ITER .LT. 15) GO TO 900
      WRITE(6,200) T,H,HT
  200 FORMAT(39H1TEMPERATURE-ENTHALPY DID NOT CONVERGE /3E15.6)
C     CALL OUTPUT(4)
      STOP
  999 CONTINUE
      TO = T
C
 1000 CONTINUE
C ENTROPY PER INITIAL MOL OF AIR-DIMENSIONLESS
      D1=EQ1+E1+1.
      D2=EQ2+ET+1.
      D3=EQ3+E3+1.
      D4=EQ4+E4+1.
      D5=EQ5+E5+1.
      D6=EQ6+E6+1.
      D7=EQ7+E7+1.
C TOTAL ENTROPY
      SR=Z*( X1*D1+X2*D2+X3*D3+X4*D4+X5*D5+X6*D6+X7*D7 )-Z*(X1*ALOG(X1) +
     1X2*ALOG(X2)+X3*ALOG(X3)+X4*ALOG(X4)+X5*ALOG(X5)+X6*ALOG(X6)+X7*
```

```
     2ALOG(X7))-Z*ALOG(P)
C ENTROPY PER INITIAL MOL OF AIR (S) IN BTU/LB-DEG R
      S    =SR*0.0686
C SPECIFIC HEAT AT CONSTANT VOLUME-CV
      FF1=3.+2.*AA1+AA2
      CV1=2.5+((2.*AA1*A1*A1+AA2*A2*A2)/FF1)-(((2.*AA1*A1+AA2*A2)/FF1)**
     12.)+((.25*A3*A3)/(((2.*A19)/(1.-A19*A19))**2))
      CV2=2.5+((.25*A4*A4)/(((2.*A21)/(1.-A21*A21))**2))
      CV3=1.5+((3.*AA5*A5*A5+AA6*A6*A6+5.*AA7*A7*A7+AA8*A8*A8)/(5.+3.*AA
     15+AA6+5.*AA7+AA8))-((E3-1.5)**2.)
      CV4=1.5+((10.*AA9*A9*A9+6.*AA10*A10*A10)/(4.+10.*AA9+6.*AA10))
     1-((E4-1.5)**2.)
      CV5=1.5+((10.*AA11*A11*A11+6.*AA12*A12*A12)/(4.+10.*AA11+6.*AA12))
     1-((E5-1.5)**2.)
      CV6=1.5+((3.*AA13*A13*A13+5.*AA14*A14*A14+5.*AA15*A15*A15+AA16*A16
     1*A16+5.*AA17*A17**2)/(1.+3.*AA13+5.*AA14+5.*AA15+AA16+5.*AA17))-((
     2E6-1.5)**2.)
      CV7=1.5
C LOGARITHMIC DERIVATIVES
      CK1=TT*(59000./T+2.*E3-E1)
      CK2=TT*(113200./T+2.*E4-ET)
      CK3=TT*(158000./T+E5+E7-E3)
      CK4=TT*(168800./T+E6+E7-E4)
      CK34=.2*CK3+.8*CK4
      PK1= CK1+TT
      PK2= CK2+TT
      PK3= CK3+TT
      PK4= CK4+TT
      PK34=0.2*PK3+0.8*PK4
C PARTIAL DERIVATIVES REQUIRED FOR CP
      DE1P=(PK1*EE1*(1.+EE1)*(.2-EE1))/(.8*(.5-EE1))
      DE2P=(PK2*EE2*(1.2+EE2)*(.8-EE2))/(.4*(4.8-EE2))
      DE3P=.5*PK34*EE3*(1.-EE3**2)
      DZX1P=-DE1P
      DZX2P=-DE2P
```

```
      DZX3P=2.*DE1P - .4*DE3P
      DZX4P=2.*DE2P-1.6*DE3P
      DZX5P=.4*DE3P
      DZX6P=1.6*DE3P
      DZX7P=2.* DE3P
C EQUATION FOR SPECIFIC HEAT AT CONSTANT PRESSURE
      CPF=Z*(X1*(CV1+1.)+X2*(CV2+1.)+X3*(CV3+1.)+X4*(CV4+1.)+X5*(CV5+1.
     1)+X6*(CV6+1.)+X7*(CV7+1.))
      CPR = CPF +              T*(DZX1P*(EN1+1.)+DZX2P*(EN2+1.)+DZX3P*(E
     3N3+1.)+DZX4P*(EN4+1.)+DZX5P*(EN5+1.)+DZX6P*(EN6+1.)+DZX7P*(EN7+1.)
     4)
      CPF = CPR
C SPECIFIC HEAT AT CONSTANT  PRESSURE (CP) IN BTU/LB-DEG R
      CP=CPR*.0686
      CPF = CPF*.0686
C DENSITY (DEN) IN LB/FT**3
      DEN=22.03703*P/(Z*T)
C **TRANSPORT PROPERTIES**
C COLLISION CROSS SECTIONS
      S2=31.4*1.E-16*(1.+(112./T))
      SI2=(S2/3.1415927)**.5
      SI4=(1.11676-(.01490* ALOG(1.-(1.-A23) **.5))-(.23654* ALOG
     1(1.-(1.-A24)**.5))-(.11582* ALOG(1.-(1.-A25)**.5)))*1.0E-8
      S4=3.1415927*(SI4)**2
      SI24=(SI2+SI4)/2.
      S24=3.1415927*(SI24)**2
      S47=9.40*1.0E-14/TSQRT
      FI=ALOG(1.042*1.0E-7*TSQ*(P*X7 )**(-.5))
      S7=8.55644*1.0E-6*(1./TSQ)*FI
      SIP4=(1.11676-(.0149*ALOG(1.-(1.-2.*A23)**.5))-(.23654*ALOG(1.-(1.
     1-2.*A24)**.5))-(.11582* ALOG(1.-(1.-2.*A25)**.5)))*1.0E-8
      SP4=3.145927*(SIP4)**2
      SIP24=(SI2+SIP4)/2.
      SP24=3.145927*SIP24**2
C COMPONENT MOL FRACTIONS FOR INDEPENDENT REACTIONS
```

```
      F1=1.+EE1
      F2=1.2+EE2
      F3=1.+EE3
      X1OD=(.2-EE1)/F1
      X2OD=.8/F1
      X3OD=2.*EE1/F1
      X2ND=(.8-EE2)/F2
      X3ND=.4/F2
      X4ND=2.*EE2/F2
      X4I=(1.-EE3)/F3
      X6I=EE3/F3
C MEAN FREE PATH RATIOS
      SS1=S24/S2
      SS2=S4/S2
      SS3=S7/S2
      SS4=S47/S2
      FP1OD=X1OD+X2OD*.9660918 +X3OD*SS1*.8164966
      FP2OD=X1OD*1.032796+X2OD+X3OD*SS1*.8528029
      FP3OD=X1OD*1.154701*SS1+X2ND*SS1*1.128152+X3OD*SS2
      FP2ND=X2ND+X4ND*SS1*.8164966+X3ND*SS1*.8528029
      FP3ND=X2ND*SS1*1.128152+X4ND*SS2*.9660918+X3ND*SS2
      FP4ND=X2ND*SS1*1.154701+X4ND*SS2+X3ND*SS2*1.032796
      FP4I=X4I*SS2+X6I*SS2
      FP6I=X4I*SS2+X6I*SS3
      FP7I=X4I*SS4*1.414186+X6I*SS3*1.414186+X6I*SS3
C VISCOSTITES OF THE COMPONENTS FOR THE DIFFERENT REACTIONS
      V1OD=1.054093*X1OD*1./FP1OD
      V2OD=.9860133*X2OD*1./FP2OD
      V3OD=.745356*X3OD*1./FP3OD
      V2ND=.9860133*X2ND*1./FP2ND
      V3ND=.745356*X3ND*1./FP3ND
      V4ND=.6972167*X4ND*1./FP4ND
      V4I=.6972167*X4I*1./FP4I
      V6I=.6972167*X6I*1./FP6I
      V7I=.4367848*1.0E-2*X6I*1./FP7I
```

```
      VROD=V1OD+V2OD+V3OD
      VRND=V2ND+V3ND+V4ND
      VRI=V4I+V6I+V7I
      F4=EE2/(.2-EE1+EE2)
      F5=2.*EE3/(.8-EE2+2.*EE3)
      VR=VROD+(F4*(VRND-VROD))+(F5*(VRI-VRND))
C TOTAL VISCOSITY (V) IN LB/FT-SEC
      V=VR*.9841838*1.0E-6*TSQRT/(1.+A22)
C CONDUCTIVITY DUE TO MOLECULAR COLLISIONS FOR DIFFERENT REACTIONS
      G1=.2105263*CV1+.4736842
      G2=.2105263*CV2+.4736842
      G3=.2105263*CV3+.4736842
      G4=.2105363*CV4+.4736842
      G5=.2105363*CV6+.4736842
      G6=.2105363*CV7+.4736842
      XKNOD=(V1OD*.9*G1)+(V2OD*1.028571*G2)+(V3OD*1.8*G3)
      XKNND=(V2ND*1.028571*G2)+(V3ND*1.8*G3)+(V4ND*2.057143*G4)
      XKNI=(V4I*2.057143*G4)+(V6I*2.057143*G5)+(V7I*52416.0*G6)
      XKN=XKNOD+(F4*(XKNND-XKNOD))+(F5*(XKNI-XKNND))
C CONDUCTIVITY DUE TO CHECMICAL REACTIONS FOR THE DIFFERENT REACTIONS
      XKROD=(.178637*(T*PK1)**2)/((SP24/(1.732051*S2))*(((X3OD+2.*X1OD)
     1**2)/(X3OD*X1OD)+(4.*X2OD/X3OD))+(X2OD/(1.414214*X1OD)))
      XKRND=(.178637*(T*PK2)**2)/((SP24/(1.732051*S2))*((((X4ND+2.*X2ND)
     1**2)/(X4ND*X2ND))+(X3ND/X2ND)) +(SP4*2.*X3ND/(S2*X4ND)))
      XKR1=(.178637*(T*PK34)**2)/(((.5*SP4/S2)+(.4347826*1.0E-2*S47/S2))
     1*((X4I+X6I)**2)/(X4I*X6I))
      XKOD=XKNOD+XKROD
      XKND=XKNND+XKRND
      XKI=XKNI+XKR1
      XKR=XKOD+(F4*(XKND-XKOD))+(F5*(XKI-XKND))
C TOTAL THERMAL CONDUCTIVITY (XK) IN BTU/FT-SEC-DEG R
      XK=XKR*((.3206522*1.0E-6*TSQRT)/(1.+A22))
C PRANDTL NUMBER (PR) DIMENSIONLESS
      PRN = .2105263 * CPR * VR / XKR
      IF(I.EQ. 1) PRW = PRN
```

```
C FORM REQUIRED BY CALL STATEMENT
C
C      **  RHO UNITS SLUGS/FT**3
C      **  MU UNITS LBM/FT-SEC
C      **  RM UNITS LBF**2 SEC**3/FT**6
C
      MU (I) = V
      RHO(I)=DEN/32.174
      RM(I)=RHO(I)*MU(I )/32.174
      AK(I) = XK
      CPT(I) = CPF
C *** CALCULATE  THE MEAN MOLECULAR WT. ***
      REAL = 25050.*S    *Z / SR
      AMW(I)= GASC / REAL
C MASS FRACTIONS
      C(1,I) = X1    *32.00/AMW(I)
      C(2,I) = X2    *28.00/AMW(I)
      C(3,I) = X3    *16.00/AMW(I)
      C(4,I) = X4    *14.00/AMW(I)
      C(6,I) = X6    *14.00/AMW(I)
      C(5,I) = X5    *16.00/AMW(I)
      C(7,I) = X7    /(1820.*AMW(I))
C SPECIES ENTHALPY PER INITIAL MCLE OF AIR IN BTU/LB OF I
      HS(1,I) = (Z*X1*EN1/C(1,I) +Z)*T*.12348
      HS(2,I) = (Z*X2*EN2/C(2,I) +Z)*T*.12348
      HS(3,I) = (Z*X3*EN3/C(3,I) +Z)*T*.12348
      HS(4,I) = (Z*X4*EN4/C(4,I) +Z)*T*.12348
      HS(5,I) = (Z*X5*EN5/C(5,I) +Z)*T*.12348
      HS(6,I) = (Z*X6*EN6/C(6,I) +Z)*T*.12348
      HS(7,I) = (Z*X7*EN7/C(7,I) +Z)*T*.12348
 2000 CONTINUE
      MUDZ= MU(NETA)
      RDZ= RHO(NETA)
      RMDZ= RM(NETA)
C
```

```
C
      DO 40 I=KODE,NETA
C
C     **  NONDIMENSIONALIZE  RHO  AND  MU  **
C
      RHO(I) = RHO(I)/RDZ
      MU(I) = MU(I)/MUDZ
      RM(I) =RM(I)/RMDZ
      AK(I) = AK(I)*AKNF
      CPT(I) = CPT(I)*CPNF
C NONDIMENSIONAL SPECIES ENTHALPY
      HS(1,I) = HS(1,I)*HNF
      HS(2,I) = HS(2,I)*HNF
      HS(3,I) = HS(3,I)*HNF
      HS(4,I) = HS(4,I)*HNF
      HS(5,I) = HS(5,I)*HNF
      HS(6,I) = HS(6,I)*HNF
      HS(7,I) = HS(7,I)*HNF
C
C
 40   CONTINUE
  100 FORMAT(1X,9E14.6)
      RETURN
      END
```

```
      SUBROUTINE FG2 (Y,RHOA,TN,K)
C--------ORDER OF SPECIES ISO
C        CO         C          C2         C3         C+         C2H        CN
C         1         2           3          4          5          6          7
C       HCN        H2           H         N2          N         N+          O
C         8         9          10         11         12         13         14
C        O+        E-        C2H2        C3H        C4H
C        15        16          17         18         19
      COMMON/FOG/AII     ,B     ,C    ,            T
      DIMENSION Y(19),T1(5)
      DIMENSION RHO(2),QK(16,2),PL(16,10), RATES(3,16),EQK(16)
      DATA R/82.05/
      DATA RATES/
     1    .4352597E 02,   .5000000E 00,   .3572848E 05,   .2683251E 02,
     2    .5000000E 00,   .1761263E 05,   .4835429E 02,  -.1500000E 01,
     3    .1131738E 06,   .4272747E 02,  -.8200000E 00,   .5193211E 05,
     4    .2866064E 02,   .5000000E 00,   .1575073E 06,   .2870602E 02,
     5    .5000000E 00,   .1675716E 06,   .4684016E 02,  -.1000000E 01,
     6    .6632414E 05,   .4357476E 02,   .5000000E 00,   .6038617E 05,
     7    .4369782E 02,   .5000000E 00,   .7045053E 05,   .4369782E 02,
     8    .5000000E 00,   .5887652E 05,   .4588918E 02,  -.1000000E 01,
     9    .1297799E 06,   .4374912E 02,   .5000000E 00,   .9561144E 05,
     A    .4367655E 02,   .5000000E 00,   .7799880E 05,   .4368724E 02,
     1    .5000000E 00,   .1333528E 06,   .4369782E 02,   .5000000E 00,
     2    .8303098E 05,   .4369782E 02,   .5000000E 00,   .7296662E 05/
      DIMENSION E1(95),E2(95),E3(34),E(16,7,2)
      EQUIVALENCE (E(1,1,1),E1(1))
      DATA E1/
     1 0.3530E 00, 0.1816E 01, 0.1727E 01, 0.1642E 01, 0.2321E 01,
     2 0.2753E 01, 0.1675E 01, 0.2257E 01, 0.3458E 01, 0.2094E 01,
     3 0.2028E 01, 0.3053E 01, 0.7810E 00, 0.2497E 01, 0.2132E 01,
     4 0.7030E 00,-0.7681E-03,-0.3571E-02,-0.8059E-03,-0.2810E-03,
     5 0.2250E-03,-0.3726E-03,-0.6017E-03,-0.2955E-02,-0.3852E-02,
     6-0.2155E-02,-0.1370E-02,-0.4032E-02,-0.1175E-03, 0.6404E-04,
     7-0.2840E-02,-0.1406E-02, 0.5153E-06, 0.1634E-05, 0.1344E-06,
```

```
8-0.9244E-07,-0.9094E-07, 0.1884E-06,-0.6914E-07, 0.1359E-05,
9 0.1541E-05, 0.7206E-06, 0.4461E-06, 0.1731E-05,-0.8460E-07,
A-0.5011E-07, 0.1138E-05, 0.5735E-06,-0.9618E-10,-0.2783E-09,
1 0.4980E-11, 0.2914E-10, 0.1275E-10,-0.3765E-10, 0.3996E-10,
2-0.2518E-09,-0.2492E-09,-0.1067E-09,-0.5622E-10,-0.2858E-09,
3 0.2854E-10, 0.6619E-11,-0.2020E-09,-0.8895E-10, 0.5427E-14,
4 0.1617E-13,-0.9780E-15,-0.2112E-14,-0.5056E-15, 0.2271E-14,
5-0.2991E-14, 0.1668E-13, 0.1406E-13, 0.5735E-14, 0.2436E-14,
6 0.1621E-13,-0.2197E-14,-0.2271E-15, 0.1335E-13, 0.4609E-14,
7 0.3485E 05, 0.1329E 05, 0.1132E 06, 0.5196E 05, 0.1580E 06,
8 0.1686E 06, 0.9406E 05, 0.5850E 05, 0.6525E 05, 0.5766E 05,
9 0.1289E 06, 0.8906E 05, 0.7297E 05, 0.1306E 06, 0.8068E 05/
 EQUIVALENCE (E(16,6,1),E2(1))
 DATA E2/
1 0.7220E 05, 0.7500E-01,-0.8961E 01, 0.4274E 01, 0.2626E 01,
2-0.1181E 02,-0.1238E 02, 0.3698E 01, 0.1912E 01,-0.6335E 01,
3 0.3671E 01, 0.3773E 01, 0.1034E 01, 0.9378E 01,-0.1216E 02,
4 0.5461E 01, 0.1162E 02,-0.9100E-01,-0.1852E 01, 0.1765E 01,
5 0.4505E 01, 0.2904E 01, 0.2261E 01, 0.1414E 01, 0.3753E 01,
6 0.2653E 01, 0.2452E 01, 0.1323E 01,-0.1596E 02, 0.2560E 00,
7 0.2895E 01, 0.3483E 01, 0.2320E 00, 0.2624E-03, 0.1831E-02,
8-0.1250E-02,-0.4018E-02,-0.3576E-03, 0.3808E-03,-0.8027E-03,
9-0.4486E-02,-0.2187E-02,-0.2382E-02,-0.5403E-03, 0.1840E-01,
A 0.1581E-03,-0.3234E-03,-0.4981E-02,-0.8811E-03,-0.8699E-09,
1-0.5850E-06, 0.3816E-06, 0.1254E-05, 0.6591E-07,-0.1210E-06,
2 0.1697E-06, 0.1768E-05, 0.6688E-06, 0.6930E-06, 0.1688E-06,
3-0.5690E-05,-0.3970E-07, 0.4932E-07, 0.2072E-05, 0.4090E-06,
4-0.5441E-11, 0.7641E-10,-0.3536E-10,-0.1592E-09,-0.3159E-11,
5 0.1070E-10,-0.1315E-10,-0.2902E-09,-0.8277E-10,-0.8428E-10,
6-0.1859E-10, 0.6841E-09, 0.2542E-11,-0.2579E-11,-0.3588E-09,
7-0.6840E-10, 0.2100E-15,-0.3417E-14, 0.1003E-14, 0.7013E-14,
8 0.2813E-16,-0.3032E-15, 0.3831E-15, 0.1768E-13, 0.3596E-14,
9 0.3630E-14, 0.5931E-15,-0.2842E-13, 0.2920E-17, 0.2381E-16/
 EQUIVALENCE (E(15,5,2),E3(1))
 DATA E3/
```

```
     1 0.2234E-13, 0.3694E-14, 0.4160E 05, 0.1329E 05, 0.1132E 06,
     2 0.5196E 05, 0.1580E 06, 0.1686E 06, 0.8731E 05, 0.6525E 05,
     3 0.6525E 05, 0.5766E 05, 0.1289E 06, 0.8906E 05, 0.7297E 05,
     4 0.1306E 06, 0.8068E 05, 0.7220E 05, 0.2066E 01, 0.1126E 02,
     5 0.4450E 01,-0.1348E 02,-0.1538E 02,-0.1000E 02, 0.5594E 01,
     6-0.6819E 01,-0.2220E 01, 0.1504E 01, 0.7660E 01, 0.1101E 03,
     7 0.1266E 02,-0.1481E 02,-0.1881E 01, 0.1435E 02/
C         RHOA= DENSITY (LBM/FT3)
      RHOB= .031085*RHOA
      RHO(1)=.515362*RHOB
      RHO(2)=RHO(1)*RHO(1)
      X=0.
      DO 6 I=1, 19
    6 X=X+Y(I)
      IF(ABS((TN-T)/TN)-.0001)20,20,10
   10 T=TN
      T1(1) = T
      DO12M=2,4
   12 T1(M) = T1(1)*T1(M-1)
      T1(5) = ALOG(T1(1))
      RT  = R*T
      L=1
      IF(T1(1).GT.6000.)L=2
      DO 18 I=1, 16
      QK(I,1)=T**RATES(2,I)* EXP(RATES(1,I)-RATES(3,I)/T)
      QK(I,2) = E(I,1,L)*(1.-T1(5))-E(I,2,L)*T1(1)/2.-E(I,3,L)*T1(2)/6.
     X-E(I,4,L)*T1(3)/12.-E(I,5,L)*T1(4)/20.+E(I,6,L)/T1(1)-E(I,7,L)
      EQK(I) = EXP(-QK(I,2))
      QK(I,2) = QK(I,1)/ EQK(I)
      IF(I.GT.2)QK(I,2) = RT*QK(I,2)
   18 CONTINUE
   20 PL(  1,  1) = QK(  1,  1)*RHO(1)*Y( 12)
      IND = 0
      PL(  1,  2) = QK(  1,  1)*RHO(1)*Y(  1)
      PL(  1,  6) = QK(  1,  2)*RHO(1)*Y( 14)
```

```
      PL(  1,  7) = QK(  1,  2)*RHO(1)*Y(  7)
      PL(  2,  1) = QK(  2,  1)*RHO(1)*Y( 10)
      PL(  2,  2) = QK(  2,  1)*RHO(1)*Y(  6)
      PL(  2,  6) = QK(  2,  2)*RHO(1)*Y(  9)
      PL(  2,  7) = QK(  2,  2)*RHO(1)*Y(  3)
      PL(  3,  1) = QK(  3,  1)*RHO(1)*X
      PL(  3,  6) = 2.*QK(  3,  2)*RHO(2)*Y( 12)*X
      PL(  3,  5) = QK(  3,  1)*RHO(1)*Y( 11)
      PL(  3, 10) = QK(  3,  2)*RHO(2)*Y( 12)*Y( 12)
      PL(  4,  1) = QK(  4,  1)*RHO(1)*X
      PL(  4,  6) = 2.*QK(  4,  2)*RHO(2)*Y( 10)*X
      PL(  4,  5) = QK(  4,  1)*RHO(1)*Y(  9)
      PL(  4, 10) = QK(  4,  2)*RHO(2)*Y( 10)*Y( 10)
      PL(  5,  1) = QK(  5,  1)*RHO(1)*X
      PL(  5,  6) = QK(  5,  2)*RHO(2)*Y( 16)*X
      PL(  5,  7) = QK(  5,  2)*RHO(2)*Y( 15)*X
      PL(  5,  5) = QK(  5,  1)*RHO(1)*Y( 14)
      PL(  5, 10) = QK(  5,  2)*RHO(2)*Y( 15)*Y( 16)
      PL(  6,  1) = QK(  6,  1)*RHO(1)*X
      PL(  6,  6) = QK(  6,  2)*RHO(2)*Y( 16)*X
      PL(  6,  7) = QK(  6,  2)*RHO(2)*Y( 13)*X
      PL(  6,  5) = QK(  6,  1)*RHO(1)*Y( 12)
      PL(  6, 10) = QK(  6,  2)*RHO(2)*Y( 13)*Y( 16)
      PL(  7,  1) = QK(  7,  1)*RHO(1)*X
      PL(  7,  6) = QK(  7,  2)*RHO(2)*Y( 12)*X
      PL(  7,  7) = QK(  7,  2)*RHO(2)*Y(  2)*X
      PL(  7,  5) = QK(  7,  1)*RHO(1)*Y(  7)
      PL(  7, 10) = QK(  7,  2)*RHO(2)*Y(  2)*Y( 12)
      PL(  8,  1) = QK(  8,  1)*RHO(1)*X
      PL(  8,  6) = QK(  8,  2)*RHO(2)*Y( 10)*X
      PL(  8,  7) = QK(  8,  2)*RHO(2)*Y(  7)*X
      PL(  8,  5) = QK(  8,  1)*RHO(1)*Y(  8)
      PL(  8, 10) = QK(  8,  2)*RHO(2)*Y(  7)*Y( 10)
      PL(  9,  1) = QK(  9,  1)*RHO(1)*X
      PL(  9,  6) = QK(  9,  2)*RHO(2)*Y( 10)*X
```

```
PL(  9,  7) = QK(  9,  2)*RHO(2)*Y(  3)*X                                        FG2 1410
PL(  9,  5) = QK(  9,  1)*RHO(1)*Y(  6)                                          FG2 1420
PL(  9, 10) = QK(  9,  2)*RHO(2)*Y(  3)*Y( 10)                                   FG2 1430
PL( 10,  1) = QK( 10,  1)*RHO(1)*X                                               FG2 1440
PL( 10,  6) = QK( 10,  2)*RHO(2)*Y( 10)*X                                        FG2 1450
PL( 10,  7) = QK( 10,  2)*RHO(2)*Y(  6)*X                                        FG2 1460
PL( 10,  5) = QK( 10,  1)*RHO(1)*Y( 17)                                          FG2 1470
PL( 10, 10) = QK( 10,  2)*RHO(2)*Y(  6)*Y( 10)                                   FG2 1480
PL( 11,  1) = QK( 11,  1)*RHO(1)*X                                               FG2 1490
PL( 11,  6) = QK( 11,  2)*RHO(2)*Y( 14)*X                                        FG2 1500
PL( 11,  7) = QK( 11,  2)*RHO(2)*Y(  2)*X                                        FG2 1510
PL( 11,  5) = QK( 11,  1)*RHO(1)*Y(  1)                                          FG2 1520
PL( 11, 10) = QK( 11,  2)*RHO(2)*Y(  2)*Y( 14)                                   FG2 1530
PL( 12,  1) = QK( 12,  1)*RHO(1)*X                                               FG2 1540
PL( 12,  6) = QK( 12,  2)*RHO(2)*Y(  2)*X                                        FG2 1550
PL( 12,  7) = QK( 12,  2)*RHO(2)*Y(  3)*X                                        FG2 1560
PL( 12,  5) = QK( 12,  1)*RHO(1)*Y(  4)                                          FG2 1570
PL( 12, 10) = QK( 12,  2)*RHO(2)*Y(  3)*Y(  2)                                   FG2 1580
PL( 13,  1) = QK( 13,  1)*RHO(1)*X                                               FG2 1590
PL( 13,  6) = 2.*QK( 13,  2)*RHO(2)*Y(  2)*X                                     FG2 1600
PL( 13,  5) = QK( 13,  1)*RHO(1)*Y(  3)                                          FG2 1610
PL( 13, 10) = QK( 13,  2)*RHO(2)*Y(  2)*Y(  2)                                   FG2 1620
PL( 14,  1) = QK( 14,  1)*RHO(1)*X                                               FG2 1630
PL( 14,  6) = QK( 14,  2)*RHO(2)*Y( 16)*X                                        FG2 1640
PL( 14,  7) = QK( 14,  2)*RHO(2)*Y(  5)*X                                        FG2 1650
PL( 14,  5) = QK( 14,  1)*RHO(1)*Y(  2)                                          FG2 1660
PL( 14, 10) = QK( 14,  2)*RHO(2)*Y(  5)*Y( 16)                                   FG2 1670
PL( 15,  1) = QK( 15,  1)*RHO(1)*X                                               FG2 1680
PL( 15,  6) = QK( 15,  2)*RHO(2)*Y( 2)*X                                         FG2 1690
PL( 15,  7) = QK( 15,  2)*RHO(2)*Y( 6)*X                                         FG2 1700
PL( 15,  5) = QK( 15,  1)*RHO(1)*Y(18)                                           FG2 1710
PL( 15, 10) = QK( 15,  2)*RHO(2)*Y( 6)*Y( 2)                                     FG2 1720
PL( 16,  1) = QK( 16,  1)*RHO(1)*X                                               FG2 1730
PL( 16,  6) = QK( 16,  2)*RHO(2)*Y( 2)*X                                         FG2 1740
PL( 16,  7) = QK( 16,  2)*RHO(2)*Y(18)*X                                         FG2 1750
```

```
      PL( 16,  5) = QK( 16,  1)*RHO(1)*Y(19)
      PL( 16, 10) = QK( 16,  2)*RHO(2)*Y(18)*Y( 2)
      GO TO (31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49),K
   31 AA          = PL( 11, 10) - PL( 11,  5)
      AII      = AA        -PL(1,1)- PL(11,1)
      GO TO 90
   32 BB=    PL(  7,  6)+    PL(  7, 10)+    PL( 11,  6)+    PL( 11, 10)
      BB=BB+    PL( 12,  7)+    PL( 12, 10)+2.*PL( 13,  6)+2.*PL( 13, 10)
      AA=    PL(  7,  5)+    PL( 11,  5)+    PL( 12,  5)+2.*PL( 13,  5)
      AA=AA+    PL( 14, 10) + PL(15, 5) + PL(16, 5)
      BB=BB+    PL( 14,  1)+    PL( 14,  5) + PL(15,10) + PL(15, 7)
      BB=BB+    PL( 16,10) + PL(16, 7)
      AII         = AA - BB
      GO TO 90
   33 AA=    PL(  9,  5)+    PL( 12,  5)+    PL( 13, 10)
      BB=    PL(  9, 10)+    PL( 12, 10)+    PL( 13,  5)
      AA          = AA - BB
      AII         = AA          - PL(  2,  6) - PL(  9,  6) - PL( 12,  6)
     1 - PL( 13,  1)
      GO TO 90
   34 AA=    PL( 12, 10)
      BB=    PL( 12,  5)
      AA          = AA - BB
      AII         = AA          - PL( 12,  1)
      GO TO 90
   35 BB=    PL( 14, 10)
      AA=    PL( 14,  5)
      AA          = AA - BB
      AII         = AA          - PL( 14,  6)
      GO TO 90
   36 AA=    PL(  9, 10)+    PL( 10,  5) + PL( 15,  5)
      BB=    PL(  9,  5)+    PL( 10, 10) + PL( 15, 10) + PL( 15,  6)
      AA          = AA - BB
      AII         = AA          - PL(  2,  1) - PL(  9,  1) - PL( 10,  6)
      GO TO 90
```

```
37 AA=    PL(  7, 10)+    PL(  8,  5)
   BB=    PL(  7,  5)+    PL(  8, 10)
   AA           = AA - BB
   AII          = AA          - PL(  1,  6) - PL(  7,  1) - PL(  8,  6)
   GO TO 90
38 AA=    PL(  8, 10)
   BB=    PL(  8,  5)
   AA           = AA - BB
   AII          = AA          - PL(  8,  1)
   GO TO 90
39 AA=    PL(  4, 10)
   BB=    PL(  4,  5)
   AA           = AA - BB
   AII          = AA          - PL(  2,  7) - PL(  4,  1)
   GO TO 90
40 BB=    PL(  2,  2)+2.*PL(  4,  6)+2.*PL(  4, 10)+   PL(  8,  7)
   BB=BB+    PL(  8, 10)+    PL(  9,  7)+    PL(  9, 10)+    PL( 10,  7)
   AA=2.*PL(  4,  5)+    PL(  8,  5)+    PL(  9,  5)+    PL( 10,  5)
   BB=BB+    PL( 10, 10)
   AII          = AA - BB
   GO TO 90
41 AA=    PL(  3, 10)
   BB=    PL(  3,  5)
   AA           = AA - BB
   AII          = AA          - PL(  3,  1)
   GO TO 90
42 BB=    PL(  1,  2)+2.*PL(  3,  6)+2.*PL(  3, 10)+   PL(  6,  1)
   AA=2.*PL(  3,  5)+    PL(  6, 10)+    PL(  7,  5)
   BB=BB+    PL(  6,  5)+    PL(  7,  7)+    PL(  7, 10)
   AII          = AA - BB
   GO TO 90
43 AA=    PL(  6,  5)
   BB=    PL(  6, 10)
   AA           = AA - BB
   AII          = AA          - PL(  6,  6)
```

```
      GO TO 90
   44 AA=    PL(  5, 10)+   PL( 11,  5)
      BB=    PL(  5,  5)+   PL( 11, 10)
      AA           = AA - BB
      AII          = AA           - PL(  1,  7) - PL(  5,  1) - PL( 11,  7)
      GO TO 90
   45 AA=    PL(  5,  5)
      BB=    PL(  5, 10)
      AA           = AA - BB
      AII          = AA           - PL(  5,  6)
      GO TO 90
   46 AA=    PL(  5,  5)+   PL(  6,  5)+   PL( 14,  5)
      BB=    PL(  5, 10)+   PL(  6, 10)+   PL( 14, 10)
      AA           = AA - BB
      AII          = AA           - PL(  5,  7) - PL(  6,  7) - PL( 14,  7)
      GO TO 90
   47 AA           = PL( 10, 10) - PL( 10,  5)
      AII          = AA           - PL( 10,  1)
      GO TO 90
   48 AA       = PL(15,10) - PL(15, 5) + PL(16, 5) - PL(16,10)
      AII     =AA       - PL(15, 1) - PL(16, 6)
      GO TO 90
   49 AA        = PL(16,10) - PL(16, 5)
      AII       = AA       - PL(16, 1)
   90 PL(  1,  1)=Y(  1)*PL(  1,  1)
      PL(  1,  6)=Y(  7)*PL(  1,  6)
      PL(  2,  1)=Y(  6)*PL(  2,  1)
      PL(  2,  6)=Y(  3)*PL(  2,  6)
      PL(  3,  1)=Y( 11)*PL(  3,  1)
      PL(  3,  6)=Y( 12)*PL(  3,  6)/2.
      PL(  4,  1)=Y(  9)*PL(  4,  1)
      PL(  4,  6)=Y( 10)*PL(  4,  6)/2.
      PL(  5,  1)=Y( 14)*PL(  5,  1)
      PL(  5,  6)=Y( 15)*PL(  5,  6)
      PL(  6,  1)=Y( 12)*PL(  6,  1)
```

```
      PL(  6,  6)=Y( 13)*PL(  6,  6)                                    FG2 2810
      PL(  7,  1)=Y(  7)*PL(  7,  1)                                    FG2 2820
      PL(  7,  6)=Y(  2)*PL(  7,  6)                                    FG2 2830
      PL(  8,  1)=Y(  8)*PL(  8,  1)                                    FG2 2840
      PL(  8,  6)=Y(  7)*PL(  8,  6)                                    FG2 2850
      PL(  9,  1)=Y(  6)*PL(  9,  1)                                    FG2 2860
      PL(  9,  6)=Y(  3)*PL(  9,  6)                                    FG2 2870
      PL( 10,  1)=Y( 17)*PL( 10,  1)                                    FG2 2880
      PL( 10,  6)=Y(  6)*PL( 10,  6)                                    FG2 2890
      PL( 11,  1)=Y(  1)*PL( 11,  1)                                    FG2 2900
      PL( 11,  6)=Y(  2)*PL( 11,  6)                                    FG2 2910
      PL( 12,  1)=Y(  4)*PL( 12,  1)                                    FG2 2920
      PL( 12,  6)=Y(  3)*PL( 12,  6)                                    FG2 2930
      PL( 13,  1)=Y(  3)*PL( 13,  1)                                    FG2 2940
      PL( 13,  6)=Y(  2)*PL( 13,  6)/2.                                 FG2 2950
      PL( 14,  1)=Y(  2)*PL( 14,  1)                                    FG2 2960
      PL( 14,  6)=Y(  5)*PL( 14,  6)                                    FG2 2970
      PL( 15,  1)=Y( 18)*PL( 15,  1)                                    FG2 2980
      PL( 15,  6)=Y(  6)*PL( 15,  6)                                    FG2 2990
      PL( 16,  1)=Y( 19)*PL( 16,  1)                                    FG2 3000
      PL( 16,  6)=Y( 18)*PL( 16,  6)                                    FG2 3010
  191 GO TO (51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69),KFG2 3020
   51 AA=   PL(  1,  6)+   PL( 11,  6)                                  FG2 3030
      BB=   PL(  1,  1)+   PL( 11,  1)                                  FG2 3040
      B       = AA - BB                                                 FG2 3050
      GO TO 95                                                          FG2 3060
   52 BB=   PL(  7,  6)+   PL( 11,  6)+   PL( 12,  6)+2.*PL( 13,  6)    FG2 3070
      AA=   PL(  7,  1)+   PL( 11,  1)+   PL( 12,  1)+2.*PL( 13,  1)    FG2 3080
      AA=AA+   PL( 14,  6) + PL(15, 1) + PL(16, 1)                      FG2 3090
      BB=BB+   PL( 14,  1) + PL(15, 6) + PL(16,6)                       FG2 3100
      B       = AA - BB                                                 FG2 3110
      GO TO 95                                                          FG2 3120
   53 AA=   PL(  2,  1)+   PL(  9,  1)+   PL( 12,  1)+   PL( 13,  6)    FG2 3130
      BB=   PL(  2,  6)+   PL(  9,  6)+   PL( 12,  6)+   PL( 13,  1)    FG2 3140
      B       = AA - BB                                                 FG2 3150
```

```
      GO TO 95                                                          FG2 3160
   54 AA=    PL( 12,   6)                                               FG2 3170
      BB=    PL( 12,   1)                                               FG2 3180
      B        = AA - BB                                                FG2 3190
      GO TO 95                                                          FG2 3200
   55 AA=    PL( 14,   1)                                               FG2 3210
      BB=    PL( 14,   6)                                               FG2 3220
      B        = AA - BB                                                FG2 3230
      GO TO 95                                                          FG2 3240
   56 AA=    PL(  2,   6)+    PL(  9,   6)+    PL( 10,   1) + PL( 15,   1)FG2 3250
      BB=    PL(  2,   1)+    PL(  9,   1)+    PL( 10,   6) + PL( 15,   6)FG2 3260
      B        = AA - BB                                                FG2 3270
      GO TO 95                                                          FG2 3280
   57 AA=    PL(  1,   1)+    PL(  7,   6)+    PL(  8,   1)               FG2 3290
      BB=    PL(  1,   6)+    PL(  7,   1)+    PL(  8,   6)               FG2 3300
      B        = AA - BB                                                FG2 3310
      GO TO 95                                                          FG2 3320
   58 AA=    PL(  8,   6)                                               FG2 3330
      BB=    PL(  8,   1)                                               FG2 3340
      B        = AA - BB                                                FG2 3350
      GO TO 95                                                          FG2 3360
   59 AA=    PL(  2,   1)+    PL(  4,   6)                              FG2 3370
      BB=    PL(  2,   6)+    PL(  4,   1)                              FG2 3380
      B        = AA - BB                                                FG2 3390
      GO TO 95                                                          FG2 3400
   60 AA=    PL(  2,   6)+2.*PL(  4,   1)+    PL(  8,   1)+    PL(  9,   1) FG2 3410
      BB=    PL(  2,   1)+2.*PL(  4,   6)+    PL(  8,   6)+    PL(  9,   6) FG2 3420
      AA=AA+    PL( 10,   1)                                            FG2 3430
      BB=BB+    PL( 10,   6)                                            FG2 3440
      B        = AA - BB                                                FG2 3450
      GO TO 95                                                          FG2 3460
   61 AA=    PL(  3,   6)                                               FG2 3470
      BB=    PL(  3,   1)                                               FG2 3480
      B        = AA - BB                                                FG2 3490
      GO TO 95                                                          FG2 3500
```

```
 62 AA=    PL(  1,   6)+2.*PL(   3,   1)+    PL(   6,   6)+    PL(   7,   1)
    BB=    PL(  1,   1)+2.*PL(   3,   6)+    PL(   6,   1)+    PL(   7,   6)
    B         = AA - BB
    GO TO 95
 63 AA=    PL(  6,   1)
    BB=    PL(  6,   6)
    B         = AA - BB
    GO TO 95
 64 AA=    PL(  1,   1)+    PL(   5,   6)+    PL( 11,   1)
    BB=    PL(  1,   6)+    PL(   5,   1)+    PL( 11,   6)
    B         = AA - BB
    GO TO 95
 65 AA=    PL(  5,   1)
    BB=    PL(  5,   6)
    B         = AA - BB
    GO TO 95
 66 AA=    PL(  5,   1)+    PL(   6,   1)+    PL( 14,   1)
    BB=    PL(  5,   6)+    PL(   6,   6)+    PL( 14,   6)
    B         = AA - BB
    GO TO 95
 67 AA=PL(10,6)
    BB=PL(10,1)
    B         = AA - BB
    GO TO 95
 68 AA= PL(15, 6) + PL(16,1)
    BB= PL(15, 1) + PL(16, 6)
    B         = AA - BB
    GO TO 95
 69 AA= PL(16, 6)
    BB= PL(16, 1)
    B         = AA - BB
 95 IF(IND)110,101,110
101 IND = 1
    GAMMA = 0.
    DO102I=1,16
```

```
      IF(I.GT.2)GAMMA = 1.                                              FG2 3860
      Q = (RATES(2,I) + RATES(3,I)/T)/T                                 FG2 3870
      PL(I,1) = Q*PL(I,1)                                               FG2 3880
      Q = GAMMA/RT + Q - E(I,1,L)/T1(1) - E(I,2,L)/2. - E(I,3,L)*T1(1)/3FG2 3890
     X. - E(I,4,L)*T1(2)/4. - E(I,5,L)*T1(3)/5. - E(I,6,L)/T1(2)        FG2 3900
  102 PL(I,6) = Q*PL(I,6)                                               FG2 3910
      C = B                                                             FG2 3920
      GO TO 191                                                         FG2 3930
  110 RETURN                                                            FG2 3940
      END                                                               FG2 3950
```

```
      SUBROUTINE OUPPUT (IOUT)
      COMMON/KKM/ALP(    60),SVA(60)         ,SDIFF(60),SAII(60),SB(60)
      COMMON/FOG/AII      ,CWT,     B,     TOLD
      COMMCN /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      COMMCN /YL/XXX(60),YOND(60)
      COMMCN /VEL/ F(60),FC(60),Z(60),V(60)
      COMMON/PROP1/PI(60),RHO(60), T(60),BMW(60),CM(20,60),CE(5,60)
      DIMENSION  Y(20,60)
      EQUIVALENCE (CM(1),Y(1))
      DIMENSION ETA(60)
      EQUIVALENCE (NETA,NP), (PD,PRE), (YCND(1),ETA(1))
C
C         ****    PRINT RESULTS OF SPECIES ITERATIONS    ****
C
      NS = 19
      GO TO ( 1, 2),IOUT
C         OUTPUT CONVERGED RESULTS
    1 CONTINUE
   11 WRITE(6,61)
   61 FORMAT(/4X,'PT.',17X,'ETA',13X,'DENSITY',9X,'TEMPERATURE',12X,'VEL
     1OCITY'/)
      WRITE(6,71)(I,ETA(I),RHO(I),T(I),V(I),I=1,NP)
   71 FORMAT(I7,4E20.7)
C         OUTPUT INTERMEDIATE RESULTS
    2 WRITE(6,20)
   20 FORMAT(1H17X,'ETA',9X,'CO',10X,'C',9X,'C2',9X,'C3',9X,'C+',8X,'C2H
     1',9X,'CN',8X,'HCN',9X,'H2',10X,'H')
      DO30J=1,NP
   30 WRITE(6,40)ETA(J),(Y(I,J),I= 1,10)
   40 FORMAT(11F11.7)
      WRITE(6,50)
   50 FORMAT(    8X,'ETA',9X,'N2',10X,'N',9X,'N+',10X,'O',9X,'O+',9X,'E-'
     X,7X,'C2H2',8X,'C3H',8X,'C4H')
      DO60J=1,NP
```

```
60 WRITE(6,40)ETA(J),(Y(I,J),I=11,19)
   RETURN
   END
```

```
      SUBROUTINE COUPLE
C
C-----THIS SUBROUTINE SOLVES THE SPECIES AND ENERGY EQUATIONS
C
      COMMCN /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON /CONV/ FPRCT,TPRCT,DDAMP,TDAMP,PDTIL
      COMMON/CONV1/HDAMP
      COMMON /DEL/ DELTA,DTIL,DTILS
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,ECONV,DCONV,LT,IAB
      COMMCN /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CFNF
      COMMCN/NUMBER/NSP,NNS,NE,NC
      COMMON/PROP1/PI(60),RHO(60), T(60),AMW(60),C (20,60),CE(5,60)
      COMMON/PROP2/  MU(60),RM(60), AK(60)
      COMMCN/PROP3/CPS(20,60),HS(20,60),CP (60),HM(60)
      COMM CN/VECTOR/SUB(60),DIAG(60),SUP(60),B(60)
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMON/ID/SP(20),EL(5)
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      COMMCN/SP2/BR,S(20),CSHOCK(5)
      COMMON /VEL/ F(60),FC(60),Z(60),V(60)
      COMMCN/WALL/RVW,PRW,TWOLD,FLUX(20),CWALL(20),ECWALL(5)
      COMMON /YL/ETA(60),YOND(60)
      COMMCN /OLD/ TOLD(60),EOLD(60),RHOS(60)
      COMMON/DD/D(60)
      DIMENSION          RHOLD(60)
      REAL  MU,MUDZ
      LOGICAL MCONV,ECONV,DCCNV
      COMMON/NETI/NETA1,NETA2
      COMMON/KKB/RHOA,TA,VA,YA(19)
      COMMON/FOG/AII      ,BI    ,CI      ,TOLD1
      COMMCN/IT/AC
      COMMON/CK/ISN(20),MWT(19)
      REAL MWT
      DIMENSION  Y(20,60), SY(20,60)
      DIMENSION SACI(19,60)
```

```
      DIMENSION RED (60),RES(60),RET(60)
      DATA NS,SYMCH,SPER,TMCH/19,.50,.80,.01/
C
C-----INITIALIZE
C
      NETA1 = NETA - 1
      NETA2 = NETA - 2
      ECONV = .FALSE.
      ITER = 0
      DO10J=1,NETA
   10 TOLD(J) = T(J)
      AC = 8.128E-08*(TD**1.659)/(PI(1)*R*UINF)
      AR = R/UINF
C     **  COMPUTE THE  Y  COORDINATE   **
      YOND(1) = 0.0
      SUM = 0.0
      DO 39 K=2,NETA
      DETA = ETA(K) -ETA(K-1)
      SUM = SUM +DETA*(1./RHC(K) +1./RHO(K-1) )/2.0
      YOND(K) = DTIL *SUM
 39   CONTINUE
      DELTA = YOND(NETA)
      DO 49 K=1,NETA
      YOND(K) = YOND(K)/YOND(NETA)
 49   CONTINUE
      IF(IAB.NE.0)GO TO 528
      CALL SPECIE
      CALL PROPRT(NSP,1,NETA)
C     **** COMPUTE FLUX DIVERGENCE ****
  528 IF(IRAD.EQ.3.AND.ITYPE.EQ.1)CALL EFLUX
      IF(IRAD.EQ.3.AND.ITYPE.EQ.0)CALL LRAD
      TOLD1 = 0.
C        RDZ = DENSITY AT SHOCK (SLUGS OF M / FT**3)
      RHOD = RDZ*32.174
C        RHOD = DENSITY AT SHOCK (LBM OF M / FT**3)
```

```
C
C*****************************************************************************
C
C     **** CALCULATE DIFFUSION COEFFICIENT ****
  110 DO120J=1,NETA
  120 D(J) = AC*(T(J)**1.659)
  130 CALL PROPRT (NSP,1,NETA)
      DX1 = ETA(  2) - ETA(1)
      DO200J=2,NETA1
      DX  = ETA(J+1) - ETA(J)
      J1 = J - 1
      CO13 = 2.*RHO(J)*AK(J)/(DTIL*CP(J))
      CO14 = - RHO(J)*V(J)
      CO15 = (C1(DX,DX1)*RHO(J+1)*AK(J+1)/CP(J+1) + C2(DX,DX1)*RHO(J)*
     X        AK(J)/CP(J) + C3(DX,DX1)*RHO(J-1)*AK(J-1)/CP(J-1))*2./DTIL
      CON4 = 2.*RHO(J)*V(J)*V(J)*(C1(DX,DX1)*V(J+1) + C2(DX,DX1)*V(J) +
     X C3(DX,DX1)*V(J-1))
      CON5 = 2.*DTIL*E(J)/RHO(J)
      CON6 = 0.
      CON7 = 0.
      CO18 = 0.
      DO2023I=1,NSP
      CON6 = CON6 +  HS(I,J)*(C4(DX,DX1)*C(I,J+1) + C5(DX,DX1)*C(I,J) +
     X C6(DX,DX1)*C(I,J-1))
      CV   =                  (C1(DX,DX1)*C(I,J+1) + C2(DX,DX1)*C(I,J) +
     X C3(DX,DX1)*C(I,J-1))
      CON7 = CON7 +  HS(I,J)*CV
      CO18 = CO18 + CV*(C1(DX,DX1)*HS(I,J+1) + C2(DX,DX1)*HS(I,J) +
     X          C3(DX,DX1)*HS(I,J-1))
 2023 CONTINUE
      CV   = (2.*RHO(J)*AK(J)/CP(J) - RHO(J)*RHO(J)*D(J))/DTIL
      CO16 = CON6*CV
      CD = (C1(DX,DX1)*RHO(J+1)*RHO(J+1)*D(J+1) + C2(DX,DX1)*RHO(J)*
     X     RHO(J)*D(J) + C3(DX,DX1)*RHO(J-1)*RHO(J-1)*D(J-1))/DTIL
        CO17 = CON7*(CO15 -CD)
```

```
      CO18 = CO18*CV                                                    COUP1060
      SUB(J1) = CO13                                                    COUP1070
      DIAG(J1) = CO14 + CO15                                            COUP1080
      SUP(J1) =  0.                                                     COUP1090
      B(J1) = CON5 + CON4 + CO17 + CO18 + CO16                          COUP1100
  200 DX1 = DX                                                          COUP1110
      CALL SOLVE (HM(1),HM(NETA))                                       COUP1120
C-----CHECK FOR CONVERGENCE                                             COUP1130
      IF(IDEBUG.EQ.0)GO TO 249                                          COUP1140
      DO1532K1=1,NETA2                                                  COUP1150
      KK = K1+1                                                         COUP1160
 1532 WRITE(6,1533) K1,ETA(KK),HM(KK),B(K1)                             COUP1170
 1533 FORMAT(1X,I10,F15.4,2E20.6)                                       COUP1180
  249 DO250J=2,NETA1                                                    COUP1190
      SRCT =  (B(J-1) - HM(J))/HM(J)                                    COUP1200
      PRCT=ABS(SRCT)                                                    COUP1210
      IF (PRCT.GT.TPRCT) GO TO 260                                      COUP1220
  250 CONTINUE                                                          COUP1230
      GO TO 300                                                         COUP1240
  260 ITER=ITER+1                                                       COUP1250
      DO270J=2,NETA1                                                    COUP1260
      SRCT = (B(J-1) - HM(J))/HM(J)                                     COUP1270
      SAP = HDAMP*SRCT                                                  COUP1280
      IF(ABS(SAP).GT.TMCH)GO TO 1618                                    COUP1290
      ZIP = SAP                                                         COUP1300
      GO TO 1619                                                        COUP1310
 1618 ZIP = TMCH*ABS(SAP)/SAP                                           COUP1320
 1619 HM(J) = (1.+ZIP)*HM(J)                                            COUP1330
      IF(HM(J).LT.HM(1))HM(J) =1.0001*HM(1)                             COUP1340
      IF(HM(J).GT.1.0)HM(J) = .9999                                     COUP1350
  270 CONTINUE                                                          COUP1360
C                                                                       COUP1370
C.................. SOLVE FOR TEMPERATURE ...................          COUP1380
C                                                                       COUP1390
      DO818J=1,NETA                                                     COUP1400
```

```
  818 RED(J) = T(J)
      DO756J=2,NETA1
      B(J-1) = HM(J)
      ITEB = 0
  751 CALL PROPRT (NSP,J,J)
      DTT = (B(J-1) - HM(J))/CP(J)
      T(J) = T(J) + DTT
      DTT = DTT/T(J)
      ITEB = ITEB + 1
      IF(ITEB.GT.20)GO TO 752
      GO TO 754
  752 WRITE(6,753)J,DTT
  753 FORMAT(1X,'TEMPERATURE AT THE ',I3,'TH POINT DOES NOT CONVERGE',5X
     X       ,'CHANGE IN TEMPERATURE=',E20.6)
      STOP
C
  754 IF(ABS(DTT).GT..010)GO TO 751
  756 CONTINUE
      DO9023J=2,NETA1
      SRCT = (T(J) - RED(J))/RED(J)
      SAP = TDAMP*SRCT
      IF(ABS(SAP).GT.TMCH)GO TO 9021
      ZIP = SAP
      GO TO 9022
 9021 ZIP = TMCH*ABS(SAP)/SAP
 9022 T(J) = (1. + ZIP)*RED(J)
      IF(T(J).LT.T(1))T(J) = T(1)
 9023 IF(T(J).GT.1.)T(J) = 1.
      IF(IDEBUG.EQ.0)GO TO 889
      WRITE(6,817) (J,RED(J),T(J),J=1,NETA)
  817 FORMAT(1X,I5,20X,2F20.6)
C.....................................................
C
  889 IF(ITER.GE.MAXE ) GOTO 300
      IF(IAB)891,891,892
```

```
  891 CALL SPECIE                                                          COUP1760
      GO TO 110                                                            COUP1770
  892 CALL ELRAT                                                           COUP1780
      CALL CHEMEQ (1,NETA)                                                 COUP1790
      GO TO 110                                                            COUP1800
  300 CONTINUE                                                             COUP1810
      IF(ITER.LT.MAXE) ECONV=.TRUE.                                        COUP1820
      DO320J=2,NETA1                                                       COUP1830
      T(J) = T(J) + .35*(TOLD(J) - T(J))                                   COUP1840
      SRCT = (T(J) - TOLD(J))/TOLD(J)                                      COUP1850
C     IF(ABS(SRCT).GT.TMCH)T(J) = TOLD(J) + TMCH*SRCT*TOLD(J)/ABS(SRCT)   COUP1860
  320 CONTINUE                                                             COUP1870
      RETURN                                                               COUP1880
      END                                                                  COUP1890
```

```
      SUBROUTINE ELRAT
      COMMCN /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON/PROP1/PI(6C),RHO(60), T(60),AMW(60),C (20,60),CE(5,60)
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,ECONV,DCONV,LT,IAB
      COMMON/IT/AC
      COMMCN/NUMBER/NSP,NNS,NE,NC
      COMMON /YL/ETA(60),YOND(60)
      COMMCN/NETI/NETA1,NETA2
      COMMCN/VECTOR/SUB(60),DIAG(60),SUP(60),B(60)
      COMMON /DEL/ DELTA,DTIL,DTILS
      COMMON /VEL/ F(60),FC(60),Z(60),V(60)
      COMMON/DD/D(60)
C-------- SOLVE ELEMENT CCNSERVATION EQUATIONS ---------
      CE(1,NETA) = 1.E-03
      CE(2,NETA) = 1.E-03
      DO116K=1,NE
      DX1 = ETA(2) - ETA(1)
      DO106J=2,NETA1
      DX = ETA(J+1) - ETA(J)
      J1 = J - 1
      SUB(J1) = RHO(J)*RHC(J)*D(J)/(DTIL*DTIL)
      DIAG(J1) = ((C1(DX,DX1)*RHO(J+1)*RHO(J+1)*D(J+1) + C2(DX,DX1)*
     XRHO(J)*RHO(J)*D(J) + C3(DX,DX1)*RHO(J-1)*RHO(J-1)*D(J-1)) -
     XDTIL*RHO(J)*V(J))/(DTIL*DTIL)
      SUP(J1) = 0.
      B(J1) = 0.
  106 DX1 = DX
      CALL SOLVE (CE(K,   1),CE(K,NETA))
      DO116J=2,NETA1
  116 CE(K,J) = B(J-1)
      DO1753K=1,5
      DO1753J=2,NETA1
 1753 IF(CE(K,J).LT.0.     )CE(K,J) = CE(K,J+1)
      IF(IDEBUG.EQ.0)GO TO 11
      WRITE(6,117)(J,(CE(K,J),K=1,5),J=1,NETA)
```

```
117 FORMAT(1X,I5,5E20.7)                                              ELRA 360
 11 RETURN                                                            FLRA 370
    END                                                               ELRA 380
```

```
      SUBROUTINE SPECIE
C-----STAGNATION LINE SOLUTION OF SPECIES CONTINUITY EQUATION (SPC4)
C--------ORDER OF SPECIES ISO
C        CO          C          C2          C3          C+         C2H         CN
C         1          2           3           4           5           6          7
C       HCN         H2           H          N2           N          N+          O
C         8          9          10          11          12          13         14
C        O+         E-        C2H2         C3H         C4H
C        15         16          17          18          19
      COMMON/COT/A1,CS
      COMMON/N/NP1,NP2,NP3
      COMMON/KKB/RHOA,TA,VA,YA(19)
      COMMON/KKM/ALP(    60),SVA(60)          ,SDIFF(60),SAII(60),SB(60)
      COMMON/FOG/AII       ,CWT  ,B     ,TOLD
      COMMON/SAVE/SSUB(60),SDIAG(60),SSUP(60),SS(60),E(60)
      DIMENSION IDEN(19)
C*  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
      COMMON /DEL/ DELTA,DTIL,DTILS
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON/PROP1/PI(60),RHO(60), T(60),BMW(60),CM(20,60),CE(5,60)
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      COMMON /YL/XXX(60),YOND(60)
C         XXX IS ETA IN MAIN
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,IDEBUG,MCONV,ECONV,DCONV,LT,IAB
      COMMON/VECTOR/ CA(60),CB(60),CC(60),ALT(60)
      COMMON /VEL/ F(60),FC(60),Z(60),V(60)
      COMMON /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CFNF
      COMMON/NUMBER/NSP,NNS,NE,NC
      COMMON/CK/ISN(20),MWT(19)
C*  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
         REAL MWT
      DIMENSION  Y(20,60)
      COMMON/SIS/SY(20,60)
      EQUIVALENCE (CM(1),Y(1))
      DIMENSION ETA(60)
```

```
      EQUIVALENCE (NETA,NP), (PD,PRE), (YCND(1),ETA(1))
      DIMENSION SUB(60),DIAG(60),SUP(60)
      EQUIVALENCE (CA(1),SUB(1)), (CB(1),DIAG(1)), (CC(1),SUP(1))
      DATA NS,SYMCH,SPER/19,.50,.80/
      LOGICAL MCONV,ECONV,DCONV
C        RDZ = DENSITY AT SHOCK (SLUGS OF M/FT**3)
      RHOD = RDZ*32.174
C        RHOD = DENSITY AT SHOCK (LBM OF M/FT**3)
      ISIFI=0
      TOL2 = .02
      TOL1 = 0.250000*TOL2
      TOL3 = 2.*TOL2
      TOL4 = 4.*TOL2
      NID=2
      NIT=1
      IT=0
      TOLD=0.
C        NP=TOTAL NUMBER OF PROFILE POINTS
C        PRE  =STATIC PRESSURE(LBF/FT2)
      NP1=NP - 1
      NP2=NP - 2
      NP3=NP - 3
C******** DIMENSIONALIZE
      DO50I=1,NP
      RHO(I) = RHO(I)*RHOD
      T(I) = T(I)*TD
   50 V(I) = V(I)*UINF
C        ETA=Y(FT)
C        RHO = DENSITY (LBM/FT3)
C        T  =TEMPERATURE PROFILE(OK)
C        V  =VELOCITY PROFILE(FT/SEC)
C        NS=NO. OF SPECIES
C        MWT=MOLECULAR WEIGHTS
C        Y = MASS FRACTIONS
      DO111I=1,NSP
```

```
      DO111J=1,NP
  111 SY(I,J) = Y(I,J)
      DO112I=1,NSP
      DO112J=1,NP
  112 Y(I,J) = SY(ISN(I),J)
C     CALL OUPPUT (1)
      DO100I=1,NS
      DO100J=1,NP
  100 Y(I,J) = Y(I,J)/MWT(I)
C         Y = MOLES OF I/MASS OF MIXTURE
      DO1971J=1,NP
      DO1971K=1,NS
 1971 SY(K,J)=Y(K,J)
C         PATM=PRESSURE IN ATMOSPHERES
      AC=8.128E-08
      AC=AC/PRE
C--------CALCULATE DIFFUSION COEFFICIENT(DIFF)
      DO170J=1,NP
  170 SDIFF(J)= AC*(T(J)**1.659)
C         DIFF=(FT2/SEC)
C********COMPUTE PART OF THE ELEMENTS OF THE TRID MATRIX********
      DX1=ETA ( 2)-ETA ( 1)
      DO1530J=2,NP1
      DX=ETA (J+1)-ETA (J)
      SVA(J)=  -V(J) + (C1(DX,DX1)*RHO(J+1)*SDIFF(J+1) + C2(DX,DX1)*RHO(
     XJ)*SDIFF(J) + C3(DX,DX1)*RHO(J-1)*SDIFF(J-1))/RHO(J)
      A      =SDIFF(J)*C6(DX,DX1)+SVA(J)*C3(DX,DX1)
      G      =SDIFF(J)*C5(DX,DX1)+SVA(J)*C2(DX,DX1)
      C      =SDIFF(J)*C4(DX,DX1)+SVA(J)*C1(DX,DX1)
      IF(J.EQ.2)A1=A
      IF(J.EQ.NP1)CS=C
      IF(J.EQ.NP1)GO TO 1520
      SSUP(J-1)=C
 1520 SDIAG(J-1)=G
      IF(J.EQ.2)GO TO 1530
```

```
      SSUB(J-2)=A                                                       SPEC1060
 1530 DX1=DX                                                            SPEC1070
C********                                                               SPEC1080
      DO321I=1,NS                                                       SPEC1090
  321 IDEN(I)=0                                                         SPEC1100
C         ********** OVERALL ITERATION LOOP **********                  SPEC1110
  120 IT=IT+1                                                           SPEC1120
      YMCH=SYMCH                                                        SPEC1130
      PER=SPER                                                          SPEC1140
      IF(IT.LT.21)GO TO 1181                                            SPEC1150
 1175 YMCH=10.                                                          SPEC1160
 1181 CONTINUE                                                          SPEC1170
C         ********** SPECIES ITERATION LOOP **********                  SPEC1180
      DO200I= 1,NS                                                      SPEC1190
      IF(IDEN(I).GE.4)GO TO 200                                         SPEC1200
      IDEN(I)=4                                                         SPEC1210
      DO127J=2,NP1                                                      SPEC1220
      AMW=0.                                                            SPEC1230
      DO126K=1,NS                                                       SPEC1240
  126 AMW=AMW+Y(K,J)                                                    SPEC1250
      AMW=1./AMW                                                        SPEC1260
  127 RHO(J)=.7608*PRE *AMW/T(J)                                        SPEC1270
C         ********** COMPUTE REACTION RATE **********                   SPEC1280
      DO550J=2,NP1                                                      SPEC1290
      JJ = J - 1                                                        SPEC1300
      DO530L=1,NS                                                       SPEC1310
  530 YA(L)=Y(L,J)                                                      SPEC1320
      CALL FG2 (YA,RHO(J),T(J),I)                                       SPEC1330
      SAII(JJ) = AII*DELTA                                              SPEC1340
  550 SB(JJ) = B*DELTA                                                  SPEC1350
 3923 CONTINUE                                                          SPEC1360
C********                                                               SPEC1370
C******** COMPUTE ELEMENTS OF TRID. MATRIX AND SOLVE *********           SPEC1380
      DO1010J=1,NP2                                                     SPEC1390
      JJ=J+1                                                            SPEC1400
```

```
      DIAG(J)=SDIAG(J)+SAII(J)                                          SPEC1410
      IF(J.EQ.NP2)GO TO 1010                                            SPEC1420
      SUB(J)=SSUB(J)                                                    SPEC1430
      SUP(J)=SSUP(J)                                                    SPEC1440
 1010 ALT(J) = SAII(J)*Y(I,JJ) - SB(J)                                  SPEC1450
      ALT(  1)=ALT(  1) - A1*Y(I,  1)                                   SPEC1460
      ALT(NP2)=ALT(NP2) - CS*Y(I, NP)                                   SPEC1470
      CALL TRID (NP2)                                                   SPEC1480
      DO1080J=2,NP1                                                     SPEC1490
 1080 ALP(J)=ALT(J-1)                                                   SPEC1500
C********                                                               SPEC1510
      IF(IT.NE.1)GO TO 2694                                             SPEC1520
      IF(I.NE.13)GO TO 2691                                             SPEC1530
      DO2690JJ=2,NP1                                                    SPEC1540
 2690 Y(13,JJ)=ALP(JJ)                                                  SPEC1550
 2691 IF(I.NE.15)GO TO 2694                                             SPEC1560
      DO2692JJ=2,NP1                                                    SPEC1570
 2692 Y(15,JJ)=ALP(JJ)                                                  SPEC1580
 2694 CONTINUE                                                          SPEC1590
  128 DO130J=2,NP1                                                      SPEC1600
      ABD=(ALP(J)-Y(I,J))/Y(I,J)                                        SPEC1610
      AB=ABS(ABD)                                                       SPEC1620
      IF(ABS(Y(I,J)).LT.1.E-04.AND.ABS(ALP(J)).LT.1.E-04)GO TO 1344     SPEC1630
      IF(IDEN(I).LT.4)GO TO 6010                                        SPEC1640
      IF(AB.GT.TOL1)IDEN(I)=3                                           SPEC1650
 6010 IF(IDEN(I).LT.3)GO TO 6020                                        SPEC1660
      IF(AB.GT.TOL2)IDEN(I)=2                                           SPEC1670
 6020 IF(IDEN(I).LT.2)GO TO 6030                                        SPEC1680
      IF(AB.GT.TOL3)IDEN(I)=1                                           SPEC1690
 6030 IF(IDEN(I).LT.1)GO TO 6040                                        SPEC1700
      IF(AB.GT.TOL4)IDEN(I)=0                                           SPEC1710
 6040 CONTINUE                                                          SPEC1720
 1344 CONTINUE                                                          SPEC1730
      SAP=PER*ABD                                                       SPEC1740
      IF(ABS(SAP).GT.YMCH)GO TO 1618                                    SPEC1750
```

```
      ZIP=SAP
      GO TO 1619
 1618 ZIP=YMCH*ABS(SAP)/SAP
 1619 CONTINUE
      Y(I,J)=(1.+ZIP)*Y(I,J)
  130 CONTINUE
  200 CONTINUE
      IF(IT.LT.NIT)GO TO 374
      IF(IDEBUG.EQ.0)GO TO 823
      WRITE(6,1463)(I,IDEN(I),I=1,NS)
 1463 FORMAT(6(4X,I3,2X,'=',I5,4X))
  823 DO373I=1,NS
  373 IDEN(I)=0
  374 CONTINUE
C     IF(IT.NE.NID)GO TO 376
      DO375I=1,NS
      IF(IDEN(I).LT.3)GO TO 376
  375 CONTINUE
      GO TO 210
  376 IF(IT.LT.NIT)GO TO 3334
      NIT=NIT+5
      NID=NIT-4
      IF(IDEBUG.EQ.0)GO TO 3334
      WRITE(6,218)IT,YMCH
  218 FORMAT(//10X,'ITERATION NUMBER',I10,5X,'YMCH',F15.6//)
C     CALL OUTPUT (2)
 3334 CONTINUE
      GO TO 120
  210 DO193J=1,NP
      AMW = 0.
      DO192I=1,NSP
  192 AMW = AMW + Y(I,J)
  193 BMW(J) = 1./AMW
      DO191I=1,NS
      DO191J=1,NP
```

```
  191 Y(I.J) = Y(I.J)*MWT(I)                                               SPEC2110
C     CALL OUPPUT (1)                                                      SPEC2120
C******** NON-DIMENSIONALIZE                                               SPEC2130
      DO51I=1.NP                                                           SPEC2140
      RHO(I) = RHO(I)/RHOD                                                 SPEC2150
      T(I) = T(I)/TD                                                       SPEC2160
   51 V(I) = V(I)/UINF                                                     SPEC2170
      DO114I=1.NSP                                                         SPEC2180
      DO114J=1.NP                                                          SPEC2190
  114 SY(I,J) = Y(I,J)                                                     SPEC2200
      DO115I=1.NSP                                                         SPEC2210
      DO115J=1,NP                                                          SPEC2220
  115 Y(ISN(I).J) = SY(I.J)                                                SPEC2230
      RETURN                                                               SPEC2240
      END                                                                  SPEC2250
```

```
      SUBROUTINE SOLVE (Z0,Z1)
C-----THIS SUBROUTINE SOLVES THE DIFFERENTIAL EQUATION
C              A(X)Z'' + B(X)Z' + C(X)Z = D(X)
C         WITH BOUNDARY CONDITIONS Z(0)=Z0, AND Z(1)=Z1. THE NUMERICAL
C         INTEGRATION IS PERFORMED WITH NETA POINTS IN THE DOMAIN OF
C         INTEREST. THE VARIABLE COEFFICIENTS A, B, C, AND D MUST BE
C         EVALUATED AT NETA2=NETA-2 INTERIOR POINTS AND STORED IN SUB,
C         DIAG, SUP, AND B, RESPECTIVELY. THE SOLUTION IS RETURNED IN B.
      COMMON /NETI/NETA1,NETA2
      COMMON/VECTOR/SUB(60),DIAG(60),SUP(60),B(60)
      COMMON /YL/ETA(60),YOND(60)
      DX1 = ETA(   2) - ETA(1)
      DO40 J=2,NETA1
      DX  = ETA(J+1) - ETA(J)
      J1 = J - 1
      J2 = J - 2
C
      CA = SUB(J1)*C6(DX,DX1) + DIAG(J1)*C3(DX,DX1)
      CB = SUB(J1)*C5(DX,DX1) + DIAG(J1)*C2(DX,DX1) + SUP(J1)
      CC = SUB(J1)*C4(DX,DX1) + DIAG(J1)*C1(DX,DX1)
      DIAG(J1) = CB
      SUP(J1) = CC
      IF(J1.EQ.1)GO TO 20
      SUB(J2) = CA
      GO TO 40
   20 BC = CA
   40 DX1 = DX
      B(     1) = B(     1) -            BC*Z0
      B(NETA2) = B(NETA2) - SUP(NETA2)*Z1
      CALL TRID (NETA2)
      RETURN
      END
```

```
      SUBROUTINE CHEMEQ( NI,NF)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                     *
C     THIS SUBPROGRAM IS A REVISION OF A PROGRAM ORIGINALLY REPORTED  *
C     IN NASA-TN-D-5391 (AUGUST 1969).   THE PROGRAM COMPUTES FOR A   *
C     PRESSURE ARRAY,PP(N),TEMPERATURE ARRAY,TT(N), AND AN ARRAY      *
C     OF ELEMENTAL MASS FRACTIONS-CC(I,N), THE EQUILIBRIUM SPECIES    *
C     MASS FRACTIONS AT EACH POINT-N REPRESENTED BY THE GIVEN ARRAYS. *
C     THE EQUILIBRIUM COMPOSITIONS ARE STORED IN THE MATRIX CE(I,N).  *
C                                                                     *
C                                     DONALD D. ESCH                  *
C                                     LOUISIANA STATE UNIVERSITY      *
C                                     AUGUST 7, 1970                  *
C                                                                     *
C                                                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     IN = INITIAL POINT FOR EQUILIBRIUM CALCULATIONS.                *
C     NF = FINAL POINT.                                               *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      COMMON/WALL/RVW,PRW,TWOLD,FLUX(20),CWALL(20),ECWALL(5)
      COMMON/SP1/SS,TOL,NDBUG
      COMMON/EQ1/AI(20), BI(20), CI(20), DI(20), EI(20), FI(20), GI(20),
     X          AII(20),BII(20),CII(20),DII(20),EII(20),FII(20),GII(20)
      COMMON/EQ2/AA(20,5),ICODE(20)
      COMMON/THERM1/C(20),FORT(20)
      COMMON/ID/SP(20),EL(5)
      COMMON/NUMBER/NS ,NNS,MM,NC
      COMMON/WT/XMW(20),AWT(5)
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      COMMON /DY/ DYDT(20,60)
      COMMON/PROP1/PP(60),RO(60),TT(60),AMW(60),CE(20,60),CC(5,60)
      DIMENSION R( 7, 7),B( 7,1),YINT(20),FY(20),PI(7),FSUM(20),YSUM(20)
     1 ,X(20),DELT(20),XLAM(20)
      DIMENSION E(5),BB(5),Y(20)
      DIMENSION EOLD(5)
```

```
 5    CONTINUE
C
      MA=1
C
C-----INITIAL GUESS FOR EQUILIBRIUM CALCULATIONS....
C
      DO 10 I=1,NS
 10   Y(I) = CWALL(I)*AMW(NI)/XMW(I)
C
C-----COMPUTE THE SIZE OF THE MATRIX
C
      NA=MM+1
C
C     NS=NUMBER OF SPECIES....
C
C     CRIT=CRITERIA FOR CONVERGENCE.
C
      CRIT =TOL
      XBETA=CRIT
      BETA=0.
      LL=NS+1
      TOLD=0.0
C
C     THE REMAINDER OF THE PROGRAM COMPUTES EQUILIBRIUM COMPOSITIONS
C     CORRESPONDING TO THE ELEMENTAL MASS FRACTIONS IN THE CC-ARRAY
C     FROM POINT N = NI TO POINT N = NF.
C
      SUM = 0.
      DO1511I=1,MM
 1511 SUM = SUM + CC(I,1)
      DO2011I=1,MM
 2011 EOLD(I) = CC(I,1)/SUM
      DO5000N=NI,NF
      T = TT(N)*TD
      P=PP(N)
```

```
      BETOLD=0.0
      SUM=0.0
      DO15I=1,MM
 15   SUM=SUM + CC(I,N)
      DO20 I=1,MM
      IF(CC(I,N).LT.1.0E-10)CC(I,N)=1.0E-10
 20   E(I)=CC(I,N)/SUM
C
      CALL ALTERY(E,EOLD,Y,TOLD)
C
      TINCR=ABS(T-TOLD)
      IF(TINCR.LE..01)GOTO1750
 22   NT=1
      DO25 J=1,MM
      BB(J)=0.0
      DO 25  I=1,NS
 25   BB(J)=BB(J)+AA(I,J)*Y(I)
C
      CALL THERMO(T,P)
C
C-----THERMO SUBPROGRAM CALCULATES F/RT FOR EACH COMPONENT.....
C
C
C-----SET-UP THE R-MATRIX AND THE B-VECTOR....
C
 40   YBAR=0.
      DO50I=1,NS
   50 YBAR=YBAR+Y(I)
C
C     YBAR IS THE TOTAL NUMBER OF MOLES OF GAS SPECIES
C
C
C-----CALCULATE THE FREE ENERGY PARAMETER OF THE GAS SPECIES
C
      DO60I=1,NS
```

```
      FAC=Y(I)/YBAR                                                     CHEM1060
      IF(FAC.LT.1.E-73)FAC=1.E-73                                       CHEM1070
 60   FY(I)=Y(I)*(C(I)+ALOG(FAC))                                       CHEM1080
C                                                                       CHEM1090
C-----PROCEED TO CONSTRUCT THE R MATRIX                                 CHEM1100
C                                                                       CHEM1110
C-----INITIALIZE TO ZERO....                                            CHEM1120
C                                                                       CHEM1130
      DO75J=1,NA                                                        CHEM1140
      DO75K=1,NA                                                        CHEM1150
 75   R(J,K)=0.0                                                        CHEM1160
C                                                                       CHEM1170
      DO90J=1,MM                                                        CHEM1180
      DO90K=J,MM                                                        CHEM1190
      SUM=0.                                                            CHEM1200
      DO80I=1,NS                                                        CHEM1210
 80   SUM=SUM+AA(I,J)*AA(I,K)*Y(I)                                      CHEM1220
      R(J,K)=SUM                                                        CHEM1230
 90   R(K,J)=SUM                                                        CHEM1240
C                                                                       CHEM1250
      DO 103 K=1,MM                                                     CHEM1260
      SUM=0.                                                            CHEM1270
      DO 101 I=1,NS                                                     CHEM1280
  101 SUM=SUM+AA(I,K)*Y(I)                                              CHEM1290
      R(K,NA)=SUM                                                       CHEM1300
      R(NA,K)=SUM                                                       CHEM1310
  103 CONTINUE                                                          CHEM1320
C                                                                       CHEM1330
C ----PROCEED TO CALCULATE THE VECTOR B.                                CHEM1340
C                                                                       CHEM1350
      DO140J=1,MM                                                       CHEM1360
      SUM=0.                                                            CHEM1370
      DO130I=1,NS                                                       CHEM1380
 130  SUM=SUM+AA(I,J)*FY(I)                                             CHEM1390
 140  B(J,1)=SUM+BB(J)                                                  CHEM1400
```

```
C
      SUM=0.
      DO150I=1,NS
  150 SUM=SUM+FY(I)
      B(NA,1)=SUM
C
C-----MATRIX INVERSION IS CALLED TO PROVIDE THE SOLUTICN FOR
C     THE LINEARIZED EQUATIONS. THE SOLUTICN OF THE EQUATIONS
C     GIVES THE LAGRANGIAN MULTIPLIERS NEEDED TO COMPUTE THE MOLES
C     OF EACH GAS SPECIES.
C
      CALL MATINV(R,NA,B,MA,NMAX)
  156 CONTINUE
C
      DO160I=1,NA
C
C     PI(I)=LAGRANGINA MULTIPLIERS
C
  160 PI(I)=B(I,1)
      U=PI(NA)
      XBAR=U*YBAR
C
C-----COMPUTE THE MOLFS CF EACH SPECIE....
C
      DO170I=1,NS
  170 FSUM(I)=-FY(I)+(Y(I)/YBAR)*XBAR
      DO200I=1,NS
      PSUM=0.
      DO180J=1,MM
  180 PSUM=PSUM+PI(J)*AA(I,J)
      YSUM(I)=PSUM*Y(I)
  200 X(I)=FSUM(I)+YSUM(I)
C
C-----CHECK IF CONVERGENCE CRITERIA HAS BEEN MET. IF SO, GO TO 800
C
```

```
      BETA=0.0
      DO215I=1,NS
      DELT(I)=X(I)-Y(I)
  215 BETA=BETA+ABS(DELT(I))
      IF(BETA.GT.BETOLD)GOTO216
      IF(BETA.LT.XBETA)GOTO800
  216 CONTINUE
C
C-----COMPUTE THE CONVERGENCE PARAMENTER XLAMBD
C
      XLAMBD=1.
      DO210 I=1,NS
      IF(ABS(DELT(I)).LT.1.0E-20)DELT(I)=0.0
      IF(DELT(I).GE.0.)GOTO210
      IF(X(I).GT.0.)GOTO210
      XLAM(I)=-Y(I)/DELT(I)
      XLAMBD=AMIN1(XLAMBD,XLAM(I))
      XLAMBD=.99*XLAMBD
  210 CONTINUE
      XLAM1=XLAMBD
      IF(XLAM1.EQ.0.)XLAM1=1.0E-5
      DEBAR=0.
      DO220I=1,NS
  220 DEBAR=DEBAR+DELT(I)
C
C-----DETERMINE THE SIZE OF THE UNIT VECTOR XLAMBD.
C     APPLY THE CORRECTIONS TO OBTAIN A NEW SET OF ESTIMATES FOR THE
C     NEXT ITERATION. WHEN THE VALUE OF XLAMBD IS VERY SMALL SET THE
C     VALUES OF Y(I) EQUAL TO X(I) TO AVOID USING THE SAME VALUES OF
C     Y(I) AS WAS USED IN THE PREVIOUS ITERATION
C
C
C-----DETERMINE THE FREE ENERGY GRADIENT.  IF POSITIVE REDUCE XLAMBDA
C     DFDL=FREE ENERGY GRADIENT
C
```

```
  230 DFDL=0.
      DO280I=1,NS
      FAC=(Y(I)+XLAMBD*DELT(I))/(YBAR+XLAMBD*DEBAR)
      IF(FAC.LT.1.E-73)FAC=1.E-73
C260  DERP=(DELT(I)*YBAR-DEBAR*Y(I))/(YBAR+XLAMBD*DEBAR)
C280  DFDL=DFDL+DELT(I)*(C(I)+ALOG(FAC)) + DERP
  280 DFDL=DFDL+DELT(I)*(C(I)+ALOG(FAC))
      IF(DFDL.LT.0.000)GOTO300
      XLAMBD=.8*XLAMBD
      IF(XLAMBD.GT.1.0E-08)GOTO230
C
C-----THE VALUE OF XLAMBD THAT ASSURES CONVERGENCE HAS BEEN FOUND
C
  300 DO350I=1,NS
      IF(XLAMBD.GT.1.E-6)GOTO330
      IF(DFDL.LT.0.)GOTO330
      IF(XLAM1.LT.1.E-6)XLAM1=1.E-6
C
C-----CALCUALTE THE NEW COMPOSITION FOR THE NEXT ITERATION
C
      Y(I)=Y(I)+XLAM1*DELT(I)*.1
      GOTO340
  330 Y(I)=Y(I)+XLAMBD*DELT(I)
  340 IF(Y(I).LT.0.)Y(I)=1.E-73
  350 CONTINUE
C
  532 NT=NT+1
      BETOLD=BETA
      TOLD=T
C
C-----IF THE NUMBER OF ITERATIONS EXCEED 900 STOP COMPUTATIONS
C
      IF(NT.GT.900)GOTO6000
      GOTO40
  800 CONTINUE
```

```
C
C-----CONVERT Y(I) TO MOLE FRACTIONS.....
C
C     CONVERT EQUILIBRIUM MOLE FRACTIONS TO MASS FRACTIONS AND STORE
C     THESE VALUES IN THE CE-MATRIX.  AMW(N) IS THE AVERAGE MOLECULAR
C     WEIGHT AT THE POINT. N.
C
      SUMY=0.0
      DO900I=1,NS
  900 SUMY=SUMY+Y(I)
      AMW(N) = 0.0
      DO1000I=1,NS
      Y(I)=Y(I)/SUMY
 1000 AMW(N) = AMW(N) + Y(I)*XMW(I)
      DO1005I=1,NS
 1005 CE(I,N) = Y(I)*XMW(I)/AMW(N)
      GOTO1800
C
C-----IF THE TEMPERATURE CHANGE IS LESS THAN 50 DEGREES FROM LAST
C     F. E. M.  CALCULATION ASSUME CURRENT VALUES AS EQUAL TO LAST
C     VALUES...
C
 1750 DO1760I=1,NS
 1760 CE(I,N)=CE(I,N-1)
      AMW(N)=AMW(N-1)
 1800 CONTINUE
C
C     OPTIONAL OUTPUT OF POSITION , TEMPERATURE AND EQUILIBRIUM
C     COMPOSITIONS.
C
      IF(NDBUG.LT.1)GOTO3000
      PRINT 2000,P,T     ,NT
 2000 FORMAT(//,'  P = ',F5.3,' T(OK) = ',F6.0,5X,'NUMBER OF ITERATIONS
     X=',I3,/,11X,'Y(I)',12X,'C(I,N)',/)
      PRINT 2005,(SP(I),Y(I),CE(I,N),I=1,NS)
```

```
 2005 FORMAT(1X,A4,2E18.8)                                              CHEM2810
C                                                                       CHEM2820
 3000 CONTINUE                                                          CHEM2830
 3300 XBETA=CRIT                                                        CHEM2840
 5000 CONTINUE                                                          CHEM2850
 8000 CONTINUE                                                          CHEM2860
      RETURN                                                            CHEM2870
 6000 PRINT6001                                                         CHEM2880
 6001 FORMAT( ' NUMBER OF ITERATIONS EXCEEDED 900, PROGRAM TERMINATING')CHEM2890
      RETURN                                                            CHEM2900
      END                                                               CHEM2910
```

```
      SUBROUTINE ALTERY(E,EOLD,Y,TOLD)
      COMMON/WT/SMW(20),AWT(5)
      COMMON/NUMBER/NSP,NNS,NE,NC
      COMMCN/EQ2/AA(20,5),ICODE(20)
      COMMON/ELSP/LSP(5)
      COMMON /SP1/SS,TOL,NDBUG
      DIMENSION E(5),Y(20),B(5),EOLD(5)
C
      IF(NDBUG.GT.1)PRINT101
 101  FORMAT(' INITIAL GUESS CN MOLE FRACTICNS UPDATED')
C
C-----ASSUME ALL SPECIES HAVE THE SAME COMPOSITION......
C
C
C-----COMPUTE GRAM-ATOMS OF EACH ELEMENT FROM KNOWN ELEMENTAL CPMPOS
C     DISTRIBUTION AND THE MAXIMUM POSSIBLE MOLECULAR WEIGHT......
C
      DO20J=1,NE
      IF(E(J).GT.1.0E-8)GOTO20
      DO15I=1,NSP
      IF(AA(I,J).LE.0.0)GOTO15
      Y(I)=1.0E-12
 15   CONTINUE
 20   E(J)=E(J)*100./AWT(J)
C
C-----CALCULATE FOR EACH ELEMENT, THE NUMBER OF G-ATOMS BASED ON THE
C     FIRST GUESS.  ADJUST THE COMPOSITION OF EACH ELEMENT-SPECIE AS
C     REQUIRED......
C
      DO30J=1,NE
      B(J)=0.0
      DO30I=1,NSP
 30   B(J)=B(J)+AA(I,J)*Y(I)
      DO40J=1,NE
      EOLD(J) = E(J)
```

```
40    Y(LSP(J))=Y(LSP(J)) + (E(J)-B(J))                          ALTE 360
      TOLD=0.0                                                   ALTE 370
      RETURN                                                     ALTE 380
      END                                                        ALTE 390
```

```
      SUBROUTINE THERMO(T,P)
C
C
```

```
C-----SUBROUTINE THERMO CALCULATES THE FREE ENERGY FUNCTION FOR EACH
C       CHEMICAL SPECIE.
C
C
        COMMCN/NUMBER/NQ ,NNS,NE,NC
        COMMON/EQ1/AI(20), BI(20), CI(20), DI(20), EI(20), FI(20), GI(20),
       X             AII(20),BII(20),CII(20),DII(20),EII(20),FII(20),GII(20)
        COMMCN/EQ2/AA(20,5),ICODE(20)
        COMMON/THERM1/C(20),FORT(20)
C
C       T=TEMPERATURE IN OK
C
        T1=T
        T2=T1*T
        T3=T2*T
        T4=T3*T
        T5=T4*T
C
C
C-----CALCUALTE THE FREE ENERGY FUNCTICN FCRT(I)
C
C       FORT(I)=FREE ENERGY FUNCTICN
C
        DO 41 I=1,NQ
        IF(T.GT.6000.   )GOTO6205
        FORT(I)=AI(I)*(1.-ALOG(T))-BI(I)*T/2.-CI(I)*T2/6.-DI(I)*T3/12.
       1         -FI(I)*T4/20.+FI(I)/T-GI(I)
        IF(ICODE(I).EQ.1)GOTC41
        C(I)=FORT(I)+ALOG(P)
        GOTO41
   6205 FORT(I)=AII(I)*(1.-ALOG(T))-BII(I)*T/2.-CII(I)*T2/6.-DII(I)*T3/12.
       1         -EII(I)*T4/20.+FII(I)/T-GII(I)
        IF(ICODE(I).EQ.1)GOTO41
        C(I)=FORT(I)+ALOG(P)
     41 CONTINUE
```

```
      SUBROUTINE MATINV(A,N,B,M,NMAX)
C
C
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
      DIMENSION A(7,7),B(7,1),IPIVOT(7),INDEX(7,2)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX,T,SWAP)
C
C     INITIALIZATION
C
    5 ISCALE=0
    6 R1=10.0**18
    7 R2=1.0/R1
   10 DETERM=1.0
   15  DO 20 J=1,N
   20 IPIVOT(J)=0
   30 DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
   40 AMAX=0.0
   45 DO 105 J=1,N
   50 IF (IPIVOT(J)-1)60,105,60
   60 DO 100 K=1,N
   70 IF (IPIVOT(K)-1)80,100,740
   80 IF (ABS(AMAX)-ABS(A(J,K)))85,100,100
   85 IROW=J
   90 ICOLUM=K
   95 AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
      IF (AMAX)110,106,110
  106 DETERM=0.0
      ISCALE=0
      GO TO 740
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
```

```
C
C      INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130 IF (IROW-ICOLUM)140,260,140
  140 DETERM=-DETERM
  150 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
  205 IF(M)260,260,210
  210 DO 250 L=1,M
  220 SWAP=B(IROW,L)
  230 B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
  270 INDEX(I,2)=ICOLUM
  310 PIVOT=A(ICOLUM,ICOLUM)
C
C      SCALE THE DETERMINANT
C
 1000 PIVOTI=PIVOT
 1005 IF(ABS(DETERM)-R1)1030,1010,1010
 1010 DETERM=DETERM/R1
      ISCALE=ISCALE+1
      IF(ABS(DETERM)-R1)1060,1020,1020
 1020 DETERM=DETERM/R1
      ISCALE=ISCALE+1
      GO TO 1060
 1030 IF(ABS(DETERM)-R2)1040,1040,1060
 1040 DETERM=DETERM*R1
      ISCALE=ISCALE-1
      IF(ABS(DETERM)-R2)1050,1050,1060
 1050 DETERM=DETERM*R1
      ISCALE=ISCALE-1
 1060 IF(ABS(PIVOTI)-R1)1090,1070,1070
```

```
 1070 PIVOTI=PIVOTI/R1
      ISCALE=ISCALE+1
      IF(ABS(PIVOTI)-R1)320,1080,1080
 1080 PIVOTI=PIVOTI/R1
      ISCALE=ISCALE+1
      GO TO 320
 1090 IF(ABS(PIVOTI)-R2)2000,2000,320
 2000 PIVOTI=PIVOTI*R1
      ISCALE=ISCALE-1
      IF(ABS(PIVOTI)-R2)2010,2010,320
 2010 PIVOTI=PIVOTI*R1
      ISCALE=ISCALE-1
  320 DETERM=DETERM*PIVOTI
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM)=1.0
  340 DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
  355 IF(M) 380,380,360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1-ICOLUM)400,550,400
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
  455 IF(M) 550,550,460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
```

```
C                                                                       MATI1060
C     INTERCHANGE COLUMNS                                               MATI1070
  600 DO 710 I=1,N                                                      MATI1080
C                                                                       MATI1090
  610 L=N+1-I                                                           MATI1100
  620 IF(INDEX(L,1)-INDEX(L,2))630,710,630                              MATI1110
  630 JROW=INDEX(L,1)                                                   MATI1120
  640 JCOLUM=INDEX(L,2)                                                 MATI1130
  650 DO 705 K=1,N                                                      MATI1140
  660 SWAP=A(K,JROW)                                                    MATI1150
  670 A(K,JROW)=A(K,JCOLUM)                                             MATI1160
  700 A(K,JCOLUM)=SWAP                                                  MATI1170
  705 CONTINUE                                                          MATI1180
  710 CONTINUE                                                          MATI1190
  740 RETURN                                                            MATI1200
      END                                                               MATI1210
```

```
      SUBROUTINE PROPRT(NSP,NI,NF)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C        SUBROUTINE FOR THE CALCULATION OF THERMODYNAMIC AND TRANSPORT
C        PROPERTIES
C
C        NOMENCLATURE....
C              Y(I)...MOLE FRACTION OF COMPONENT I
C              C(I,N)...MASS FRACTION OF COMPONENT I AT POINT N
C              T1........TEMPERATURE, DEG. K
C              CP(I).....SPECIES SPECIFIC HEAT, CAL/GMOLE OF I-K
C              CPM(N)....MIXTURE SPECIFIC HEAT, CAL/GMOLE OF M-K
C              H(I)......SPECIES ENTHALPY, CAL/GMOLE OF I
C              HM(N).....MIXTURE ENTHALPY, CAL/GMOLE OF M
C              VIS(I)....SPECIES VISCOSITY, LBM/FT-SEC
C              VISM(N)...MIXTURE VISCOSITY, LBM/FT-SEC
C              TC(I).....SPECIES THERMAL CONDUCTIVITY, BTU/FT-SEC-DEG. R
C              TCM(N)....MIXTURE  THERMAL CONDUCTIVITY, BTU/FT-SEC-DEG. R
      COMMON /BLOCK1/V1(20),V2(20),V3(20)
      COMMON/BLOCK3/K1(20),K2(20)
      COMMON/FO1/AI(20), BI(20), CI(20), DI(20), EI(20), FI(20), GI(20),
     X           AII(20),BII(20),CII(20),DII(20),EII(20),FII(20),GII(20)
      COMMON /FRSTRM/ U INF, RINF, UINF2,RAD, RE, LXI, ITM, IEM,NETA
      COMMON/PROP1/PI(60),RHO(60),T(60),AMW(60),C (20,60),CC(5,60)
      COMMON/PROP2/VISM(60),RM(60),TCM(60)
      COMMON/PROP3/CP(20,60),H(20,60),CPM(60),HM(60)
      COMMON/WT/SMW(20),AWT(5)
      COMMON /RH/ DUD,DPHI,TO,RZB,PD,HD,HTOTAL
      COMMON /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CPNF
      DIMENSION PHI(20,20),YPHI(20),Y(20),TC(20),VIS(20)
      DIMENSION SAMW(2)
      REAL  MU,MUDZ,K1,K2
      DATA R /1.98716/
C
      DO500N=NI,NF
```

```
      T1= T(N) *TD
      T2=T1*T1
      T3=T2*T1
      T4=T3*T1
      T5=T4*T1
      DO150I=1,NSP
      IF(T1.GT.6000.)GOTO 50
      CP(I,N)=( AI(I)+ BI(I)*T1+ CI(I)*T2+ DI(I)*T3+ EI(I)*T4)*R
      H(I,N)=( AI(I)*T1+ BI(I)*T2/2.+ CI(I)*T3/3.+ DI(I)*T4/4.
     X                  + EI(I)*T5/5.+ FI(I))*R
      GOTO 100
   50 CP(I,N)=(AII(I)+BII(I)*T1+CII(I)*T2+DII(I)*T3+EII(I)*T4)*R
      H(I,N)=(AII(I)*T1+BII(I)*T2/2.+CII(I)*T3/3.+DII(I)*T4/4.
     X                  +EII(I)*T5/5.+FII(I))*R
  100 Y(I)=C(I,N)*AMW(N)/SMW(I)
      VIS(I)=(V1(I) + V2(I)*T1 + V3(I)*T2)*1.0E-05
      TC(I)=(K1(I)+K2(I)*T1)*1.0E-5
  150 CONTINUE
C-----CALCULATE PHI(I,J) PARAMETERS FOR MIXTURE PROPERTIES....
C
      DO200I=1,NSP
      DO200J=1,NSP
      VIS12=SQRT(VIS(I)/VIS(J))
      SMW14=(SMW(J)/SMW(I))**.25
      PHI(I,J)=.354*((1.+VIS12*SMW14)**2)/SQRT(1.+SMW(I)/SMW(J))
  200 CONTINUE
C
C-----CALCULATION OF MIXTURE PROPERTIES...
C
      DO250I=1,NSP
      YPHI(I)=0.0
      DO250J=1,NSP
  250 YPHI(I)=YPHI(I) + Y(J)*PHI(I,J)
      VISM(N)=0.0
      TCM(N) = 0.
```

```
      DO300I=1,NSP
      IF(Y(I).LT.1.E-5) GO TO 300
      VISM(N)=VISM(N)+Y(I)*VIS(I)/YPHI(I)
  300 TCM(N) = TCM(N) + Y(I)*TC(I)/YPHI(I)
  500 CONTINUE
C
C----- NONDIMENSIONALIZE QUANTITIES -----
C
      DO 550 N=NI,NF
      VISM(N) = VISM(N)/MUDZ
      TCM(N) = TCM(N)*AKNF
      RM(I)=RHO(N)*VISM(N)
      CPM(N) = 0.
      HM(N) = 0.
      DO 550I=1,NSP
      CP(I,N) = CP(I,N)*CPNF/SMW(I)
      H(I,N) = H(I,N)*HNF*1.80/SMW(I)
      CPM(N) = CPM(N) + CP(I,N)*C(I,N)
      HM(N) = HM(N) + H(I,N)*C(I,N)
  550 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STPSZE
C
C     **  ROUTINE TO ADJUST STEP SIZE AS NEEDED
C         TO MAINTAIN ACCURACY  **
C
C
      COMMON /DEL/ DELTA,DTIL,DTILS
      COMMCN /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON/PROP1/PI(60),RHO(60),G (60),AMW(60),C (20,60),EC(5,60)
      COMMON/PROP2/  MU(60),RM(60), AK(60)
      COMMON/PROP3/CPS(20,60),HS(20,60),CP (60),HM(60)
      COMMON /RH/ DUD,DPHI,TD,RZE,PD,HD,HTOTAL
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMCN /VEL/ F(60),FC(60),Z(60),V(60)
      COMMON/WALL/RVW,PRW,TWOLD,FLUX(20),CWALL(20),ECWALL(5)
      COMMON /YL/ETA(60),YOND(60)
      COMMON /OLD/ TOLD(60),EOLD(60),RHOS(60)
C
      N=2
  1   CONTINUE
      I2=2
      IF(NETA.GE. 59) GO TO 5
C
      DO 2 I=N,NETA
      L=I
      CHECK = ABS(G(I)-G(I-1) )
      IF(CHECK .GT. .05) GO TO 3
  2   CONTINUE
C
      GO TO 5
C
  3   CONTINUE
      M = NETA - L + 1
C
      DO 4 I=1,M
```

```
      K = NETA - I + 1
      G(K+1) = G(K)
      F(K+1) =F(K)
      DO100JJ=1,5
 100  EC(JJ,K+1)=EC(JJ,K)
      TOLD(K+1)=TOLD(K)
      RHOS(K+1)=RHOS(K)
      RHO(K+1) = RHO(K)
      RM (K+1) = RM (K)
      IF(IRAD.EQ. 3) E(K+1) = E(K)
      ETA(K+1) = ETA(K)
 4    CONTINUE
C
      G(L) = (G(L-1) + G(L+1) ) / 2.0
      F(L) = (F(L-1)+F(L+1))/2.
      DO101JJ=1,5
 101  EC(JJ,L)=(EC(JJ,L-1)+EC(JJ,L+1))/2.
      TOLD(L)=(TOLD(L-1)+TOLD(L+1))/2.0
      RHOS(L)=(RHOS(L-1)+RHOS(L+1))/2.0
      RHO(L) = (RHO(L-1) +RHO(L+1))/2.
      RM (L) = (RM (L-1) +RM (L+1))/2.
      IF(IRAD.EQ. 3) E(L) =(E(L-1)+ E(L+1))/2.0
      ETA(L) = (ETA(L-1) + ETA(L+1) )/2.0
      NETA = NETA + 1
      N=L
C
      IF( NETA .LT. 59) GO TO 1
C
 5    CONTINUE
C
      IF(I2 .GE. NETA) GO TO 10
      DO 6 I=I2,NETA,2
      L=I
      IF (L.EQ.NETA) GO TO 6
      IF(ETA(I).EQ. 0.98) GO TO 6
```

```
      CHECK =ABS(G(I+1) - G(I-1) )
      IF(CHECK .LT. 0.005) GO TO 7
 6    CONTINUE
C
      GO TO 10
 7    CONTINUE
      I2=L+1
      IF(ETA(L+1)-ETA(L-1).GT. .04) GO TO 5
C
      DO 8 I=L,NETA
      G(I) = G(I+1)
      F(I) = F(I+1)
      DO102JJ=1,5
 102  EC(JJ,I)=FC(JJ,I+1)
      TOLD(I)=TOLD(I+1)
      RHOS(I)=RHOS(I+1)
      RHO(I) = RHO(I+1)
      RM (I) = RM (I+1)
      IF(IRAD.EQ. 3) E(I) =E(I+1)
      ETA(I)=ETA(I+1)
 8    CONTINUE
C
      NETA=NETA-1
      IF (NETA .GT. 1) GO TO 5
C
 10   CONTINUE
      NN = NETA-2
      DO 20 I=1,NN
      Z(I) = ETA(I+1)/DTIL
 20   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRID (M)
C**** TRID --TRIDIAGONAL EQUATION SOLVER  OBTAINED FROM CONTE P-184 ***
C       SUBROUTINE SOLVES AX = B FOR THE VECTOR X (WHERE A IS TRIDIAGONAL
C       M = ORDER OF SYSTEM
C       SUP = SUPER DIAGONAL OF A
C       SUB = SUB DIAGONAL OF A
C       DIAG = MAIN DIAGONAL OF A
C       B = CONSTANT VECTOR
C       SUP AND DIAG ARE DESTROYED
C       SOLUTION VECTOR IS RETURNED IN B
C
      COMMON/VECTOR/ SUB(60),DIAG(60),SUP(60),B(60)
C
      N = M
      NN = N -1
      SUP(1) = SUP(1)/DIAG(1)
      B(1) = B(1)/DIAG(1)
      DO 10 I=2,N
      II = I -1
C------DECOMPOSE A TO FORM A = LU WHERE L IS LOWER TRIANGULAR,
C       AND U IS UPPER TRIANGULAR ---------
      DIAG(I) = DIAG(I) - SUP(II)*SUB(II)
      IF(I .EQ.  N) GO TO 10
      SUP(I) = SUP(I) / DIAG(I)
C------COMPUTE Z WHERE LZ = B
   10 B(I) = (B(I) - SUB(II) *B(II))/ DIAG(I)
C------COMPUTE X BY BACK SUBSTITUTION WHERE UX = Z
      DO 20 K =1,NN
      I = N - K
   20 B(I) = B(I) -SUP(I) *B(I+1)
      RETURN
      END
```

```
      FUNCTION QUAD (X,FX,I)
C
C     **  TRAPEZOIDAL QUADRATURE FUNCTION  **
C
      DIMENSION X(60),FX(60)
      DX=X(I)-X(I-1)
      QUAD = (DX/2.0) * (FX(I) + FX(I-1) )
      RETURN
      END
```

```
      SUBROUTINE EFLUX
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMON/PROP1/PI(60),RHO(60), T(60),AMW(60),C (20,60),CC(5,60)
      DIMENSION EOLD(60)
      DO 100 I=1,NETA
      IF(IEM.GT.1) EOLD(I) = E(I)
      PL = ALOG10(PI(I) )
      TSI= T(I)*TD
      TS = 1100. * PL +13800.
      IF (TS -TSI ) 300,200,200
  200 EP = 10.**(.0005 *TSI  +1.15*PL -3.15 )
      GO TO 350
  300 EP = 10.**(1.875 *PL + 3.903)
C  **** EP  HAS UNITS OF WATTS/CM**3    ***
C
  350 E(I) =(EP*R/( RINF      *UINF2 *U INF) )* 20866.0
  400 E(I) =E(I) *RZB
C      IF(IEM.GT.1) E(I) = .6 *E(I) +.4 *EOLD(I)
C
C  **** E IS NONDIMENSIONAL  ***
C
  100 CONTINUE
      RETURN
      END
```

```
FUNCTION C1 (DX,DX1)
C1 = DX1/DX/(DX+DX1)
RETURN
END
```

```
FUNCTION C2 (DX,DX1)
C2= (DX-DX1)/DX/DX1
RETURN
END
```

```
FUNCTION C3 (DX,DX1)
C3=-DX/DX1/(DX+DX1)
RETURN
END
```

```
      FUNCTION C4 (DX,DX1)
      C4=2.D0/DX/(DX+DX1)
      RETURN
      END
```

```
      FUNCTION C5 (DX,DX1)
      C5=-2.D0/DX/DX1
      RETURN
      END
```

```
      FUNCTION C6 (DX,DX1)
      C6=2.D0/DX1/(DX+DX1)
      RETURN
      END
```

```
      BLOCK DATA
      COMMON/FINV/ NHVL,NIHVC,FHVC(12),DJ(9),HVJ(9),ZKZ
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NETA
      COMMON/GUESS/TG1(60),TG2(60)
      COMMON/PROP1/PI(60),RHO(60), T(60),AMW(60),C (20,60),EC(5,60)
      COMMON/PROP2/  MU(60),RM(60), AK(60)
      COMMON/PROP3/CPS(20,60),HS(20,60),CP (60),HM(60)
      COMMON/NUMBER/NSP,NNS,NE,NC
      COMMON/ELSP/LSP(5)
      COMMON/ID/SP(20),EL(5)
      COMMON/WT/SMW(20),AWT(5)
      COMMON /BLOCK1/V1(20),V2(20),V3(20)
      COMMON/BLOCK3/K1(20),K2(20)
      COMMON/EQ1/AI(20), BI(20), CI(20), DI(20), EI(20), FI(20), GI(20),
     X          AII(20),BII(20),CII(20),DII(20),EII(20),FII(20),GII(20)
      COMMON/EQ2/AA(20,5),ICODE(20)
      COMMON/EQ3/IA(20,5)
      REAL K1,K2
      COMMON/CK/ISN(20),MWT(19)
      REAL MWT
      DATA ISN/
     1 11,    8,  19,  12,  20,  14,  13,  18,  10,    9,    2,    4,    6,
     2  3,    5,    7,  15,  16,  17,    1/
      DATA MWT/
     1 28.011, 12.011, 24.022, 36.033, 12.011, 25.030, 26.019, 27.027,
     2 02.016, 01.008, 28.016, 14.008, 14.008, 16.000, 16.000, .0005486,
     3 26.038, 37.041, 49.052/
      DATA NETA/0/
      DATA RHO /25.1,14.3,8.85,6.50,4.37,3.01,2.49,2.17,1.90,1.67,1.46,
     1         1.29,1.16,1.08,1.03,1.00,44*1.0/
      DATA RM /10.0,7.71,5.89,5.10,4.18,3.54,3.31,3.10,2.83,2.48,2.09,
     1        1.72,1.42,1.22,1.09,1.02,44*1.0/
      DATA TG1 / .1033,.2294,.3531,.4719,.5777,.6531,.6867,.7034,.7145,
     1 .7236,.7321,.7401,.7479,.7554,.7628,.7699,.7769,.7836,.7902,
     2 .7967,.8030,.8092,.8153,.8213,.8272,.8331,.8389,.8447,.8504,
```

```
3 .8562,.8619,.8676,.8734,.8791,.8850,.8908,.8968,.9028,.9089,
4 .9151,.9215,.9280,.9347,.9417,.9488,.9563,.9641,.9723,.9809,
5 .9901,10*1.0/
 DATA TG2 / .3325,.3325,.3325,.3325,.3325,.3325,.3325,.3326,.3328,
1 .3331,.3336,.3344,.3357,.3378,.3408,.3452,.3515,.3601,.3718,
2 .3873,.4076,.4335,.4665,.5075,.5560,.6054,.6487,.6857,.7161,
3 .7404,.7595,.7749,.7878,.7993,.8100,.8203,.8302,.8399,.8496,
4 .8594,.8693,.8797,.8904,.9019,.9142,.9278,.9476,.9609,.9757,
5 .9877,10*1.0/
C
 DATA NSP,NNS,NE,NC/20,0,5,20/
 DATA SP/ 'O2  ',   'N2  ',   'O   ',   'N   ',   'O+  ',
1         'N+  ',   'E-  ',   'C   ',   'H   ',   'H2  ',
2         'CO  ',   'C3-G',   'CN  ',   'C2H ',   'C2H2',
3         'C3H ',   'C4H ',   'HCN ',   'C2  ',   'C+  '/
C
 DATA SMW/32.000,   28.016,   16.000,   14.008,   16.000,
1         14.008, 5.486E-4,   12.011,    1.008,    2.016,
2         28.011,   36.033,   26.019,   25.030,   26.038,
3         37.041,   49.052,   27.027,   24.022,   12.011/
C
 DATA V1/0.1693E 01,0.9704E 00,0.1519E 01,0.2534E 00,0.0       ,
1        0.0       ,0.0       ,0.1997E 01,0.2941E 00,-.7944E-01,
2        0.2404E 01,0.2019E 01,0.2404E 01,0.2404E 01,0.1396E 01,
3        0.2019E 01,0.2019E 01,0.1378E 01,0.1931E 01,0.0       /
C
 DATA V2/0.1496E-02,0.1613E-02,0.1875E-02,0.2206E-02,0.5000E-03,
1        0.5000E-03,0.5000E-03,0.1772E-02,0.8893E-03,0.7907E-03,
2        0.1363E-02,0.1179E-02,0.1363E-02,0.1363E-02,0.8423E-03,
3        0.1179E-02,0.1179E-02,0.9651E-03,0.1393E-02,0.5000E-03/
C
 DATA V3/-.2276E-07,-.1916E-07,-.2228E-07,-.3737E-07,-.1000E-07,
1        -.1000E-07,-.1000E-07,-.3378E-07,-.8111E-08,-.8864E-08,
2        -.2184E-07,-.1655E-07,-.2184E-07,-.2184E-07,-.6939E-08,
3        -.1655E-07,-.1655E-07,-.9481E-08,-.2575E-07,-.1000E-07/
```

```
C                                                                        DATA 710
      DATA  AI/0.3316E 01,0.3221E 01,0.2670E 01,0.2474E 01,0.2491E 01,  DATA 720
     1         0.2727E 01,0.2500E 01,0.2612E 01,0.2500E 01,0.3358E 01,  DATA 730
     2         0.3254E 01,0.4002E 01,0.3411E 01,0.3485E 01,0.3891E 01,  DATA 740
     3         0.3965E 01,0.5874E 01,0.3654E 01,0.4443E 01,0.2609E 01/  DATA 750
C                                                                        DATA 760
      DATA  BI/0.1151E-02,0.5878E-03,-.1970E-03,0.9097E-04,0.2762E-04,  DATA 770
     1         -.2820E-03,0.3440E-06,-.2030E-03,-.8243E-06,0.2794E-03,  DATA 780
     2         0.9698E-03,0.3541E-02,0.4897E-03,0.3563E-02,0.5717E-02,  DATA 790
     3         0.6200E-02,0.7403E-02,0.3444E-02,-.2885E-03,-.1393E-03/  DATA 800
C                                                                        DATA 810
      DATA  CI/-.3726E-06,-.2907E-06,0.7193E-07,-.7814E-07,-.1881E-07,  DATA 820
     1         0.1105E-06,-.1954E-09,0.1095E-06,0.6421E-09,0.9372E-07,  DATA 830
     2         -.2647E-06,-.1318E-05,0.1005E-06,-.1237E-05,-.1957E-05,  DATA 840
     3         -.2265E-05,-.2729E-05,-.1258E-05,0.3036E-06,0.5959E-07/  DATA 850
C                                                                        DATA 860
      DATA  DI/0.6186E-10,0.3938E-10,-.8901E-11,0.2218E-10,0.3807E-11,  DATA 870
     1         -.1551E-10,0.3937E-13,-.1695E-10,-.1720E-12,-.2948E-10,  DATA 880
     2         0.3037E-10,0.2064E-09,-.3473E-10,0.1866E-09,0.2931E-09,  DATA 890
     3         0.3717E-09,0.4437E-09,0.2169E-09,-.6244E-10,-.1037E-10/  DATA 900
C                                                                        DATA 910
      DATA  EI/-.3666E-14,-.2000E-14,0.4002E-15,-.1489E-14,-.1028E-15,  DATA 920
     1         0.7847E-15,-.2573E-17,0.8590E-15,0.1457E-16,0.2141E-14,  DATA 930
     2         -.1177E-14,-.1144E-13,0.2361E-14,-.1013E-13,-.1585E-13,  DATA 940
     3         -.2262E-13,-.2637E-13,-.1430E-13,0.3915E-14,0.6345E-15/  DATA 950
C                                                                        DATA 960
      DATA  FI/-.1044E 04,-.1043E 04,0.2915E 05,0.5609E 05,0.1879E 06,  DATA 970
     1         0.2254E 06,-.7450E 03,0.8542E 05,0.2547E 05,-.1018E 04,  DATA 980
     2         -.1434E 05,0.9423E 05,0.4745E 05,0.5809E 05,0.2590E 05,  DATA 990
     3         0.6283E 05,0.7605E 05,0.1442E 05,0.9787E 05,0.2168E 06/  DATA1000
C                                                                        DATA1010
      DATA  GI/0.5393E 01,0.4326E 01,0.4504E 01,0.4300E 01,0.4424E 01,  DATA1020
     1         0.3645E 01,-.1173E 02,0.4144E 01,-.4612E 00,-.3548E 01,  DATA1030
     2         0.4875E 01,0.2020E 01,0.4746E 01,0.4784E 01,0.6520E 00,  DATA1040
     3         0.3467E 01,-.4010E 01,0.2373E 01,-.1090E 01,0.3709E 01/  DATA1050
```

```
C                                                                       DATA1060
      DATA AII/0.3721E 01,0.3727E 01,0.2548E 01,0.2746E 01,0.2944E 01,  DATA1070
     1         0.2499E 01,0.2508E 01,0.2141E 01,0.3934E 01,0.3363E 01,  DATA1080
     2         0.3366E 01,0.2213E 02,0.3473E 01,0.5307E 01,0.6789E 01,  DATA1090
     3         0.3965E 01,0.5874E 01,0.3654E 01,0.4026E 01,0.2528E 01/  DATA1100
C                                                                       DATA1110
      DATA BII/0.4254E-03,0.4684E-03,-.5952E-04,-.3909E-03,-.4108E-03,  DATA1120
     1         -.3725E-05,-.6332E-05,0.3219E-03,-.1776E-02,0.4656E-03,  DATA1130
     2         0.8027E-03,-.1759E-01,0.7337E-03,0.8966E-03,0.1503E-02,  DATA1140
     3         0.6200E-02,0.7403E-02,0.3444E-02,0.4857E-03,0.4869E-05/  DATA1150
C                                                                       DATA1160
      DATA CII/-.2835E-07,-.1140E-06,0.2701E-07,0.1338E-06,0.9156E-07,  DATA1170
     1         0.1147E-07,0.1364E-08,-.5498E-07,0.6013E-06,-.5127E-07,  DATA1180
     2         -.1968E-06,0.5565E-05,-.9088E-07,-.1378E-06,-.2295E-06,  DATA1190
     3         -.2265E-05,-.2729E-05,-.1258E-05,-.7026E-07,-.7026E-08/  DATA1200
C                                                                       DATA1210
      DATA DII/0.6050E-12,0.1154E-10,-.2798E-11,-.1191E-10,-.5848E-11,  DATA1220
     1         -.1102E-11,-.1094E-12,0.3604E-11,-.7819E-10,0.2802E-11,  DATA1230
     2         0.1940E-10,-.6758E-09,0.4847E-11,0.9251E-11,0.1534E-10,  DATA1240
     3         0.3717E-09,0.4437E-09,0.2169E-09,0.4666E-11,0.1134E-11/  DATA1250
C                                                                       DATA1260
      DATA FII/-.5186E-17,-.3293E-15,0.9380E-16,0.3369E-15,0.1190E-15,  DATA1270
     1         0.3078E-16,0.2934E-17,-.5564E-16,0.3482E-14,-.4905E-16,  DATA1280
     2         -.5549E-15,0.2825E-13,-.1018E-15,-.2278E-15,-.3763E-15,  DATA1290
     3         -.2262E-13,-.2637E-13,-.1430E-13,-.1142E-15,-.3476E-16/  DATA1300
C                                                                       DATA1310
      DATA FII/-.1044E 04,-.1043E 04,0.2915E 05,0.5609E 05,0.1879E 06,  DATA1320
     1         0.2254E 06,-.7450E 03,0.8542E 05,0.2547E 05,-.1018E 04,  DATA1330
     2         -.1434E 05,0.9423E 05,0.5420E 05,0.5809E 05,0.2590E 05,  DATA1340
     3         0.6283E 05,0.7605E 05,0.1442E 05,0.9787E 05,0.2168E 06/  DATA1350
C                                                                       DATA1360
      DATA GII/0.3254E 01,0.1294E 01,0.5049E 01,0.2872E 01,0.1750E 01,  DATA1370
     1         0.4950E 01,-.1208E 02,0.6874E 01,-.8598E 01,-.3716E 01,  DATA1380
     2         0.4263E 01,-.1021E 03,0.4152E 01,-.5288E 01,-.1539E 02,  DATA1390
     3         0.3467E 01,-.4010E 01,0.2373E 01,0.1090E 01,0.4139E 01/  DATA1400
```

```
C
      DATA K1/0.1019E 01,0.6541E 00,0.1250E 01,0.1281E 01,2.6000E 01,
     1        2.6000E 01,2.6000E 01,0.2506E 01,0.2496E 01,0.3211E 01,
     2        0.8589E 00,0.6304E 00,0.8589E 00,0.1126E 01,0.1126E 01,
     3        0.6304E 00,0.6304E 00,0.4855E 00,0.8589E 00,0.1000E-04/
C
      DATA K2/0.4901E-03,0.6457E-03,0.7092E-03,0.8593E-03,0.0000E-03,
     1        0.0000E-03,0.0000E-03,0.7479E-03,0.5129E-02,0.5344E-02,
     2        0.6233E-03,0.5804E-03,0.6233E-03,0.7439E-03,0.7439E-03,
     3        0.5804E-03,0.5804E-03,0.8714E-03,0.6233E-03,0.7350E-03/
C
      DATA ICODE/20*0/
C
      DATA EL/   'C',     'H',     'N',     'O',     'E',   /
      DATA AWT/ 12.011,  1.008,  14.008, 16.000, 5.486E-4  /
C
      DATA IA/ 0,0,0,0,0,0,0,1,0,0,1,3,1,2,2,3,4,1,2,1,
     H         0,0,0,0,0,0,0,0,1,2,0,0,0,1,2,1,1,1,0,0,
     N         0,2,0,1,0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,0,
     O         2,0,1,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,
     E         2,2,1,1,0,0,1,1,0,0,2,3,2,2,2,3,4,2,2,0/
C
      DATA LSP/20,9,6,5,7/
      DATA NHVL /9/, NIHVC /12/
      DATA FHVC /5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 10.8, 11.1,
     1   12.0, 13.4, 14.3, 20.0/
      DATA DJ /0.6, 2.2, 1.5, 1.65, 1.4, 1.0, 1.2, 1.4, 1.0/
      DATA HVJ /1.3, 2.7, 5.75, 7.57, 9.1, 10.4, 11.4, 12.7, 13.9/
      DATA ZKZ /7.26E-16/
      END
```

```
      SUBROUTINE LRAD
C
C
C  **  THIS IS A DRIVER PROGRAM FOR SUBROUTINE TRANS WHICH CALCULATES **
C       THE RADIATIVE FLUX DIVERGENCE THROUGH A ONE-DIMENSIONAL SLAB
C       FOR A GIVEN TEMPERATURE AND SPECIES DISTRIBUTION
      COMMON /SFLUX/ QRI(3)
      COMMON /TRN/         NUT(60), FMC(12,60), FPC(12,60),
     1               FM(9,60), FP(9,60), LINES
      COMMON /TEST/ETZ(60),IEZ
      COMMON /YL/ETA(60),YOND(60)
      COMMON/PROP1/PI(60),RHO(60), T(60),AMW(60),C (20,60),EC(5,60)
      COMMON/SIS/SY(20,60)
      COMMON /FRSTRM/ U INF, RINF, UINF2, F , RE, LXI, ITM, IEM, NETA
      COMMON /DEL/ DELTA,DTIL,DTILS
      COMMON /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CPNF
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,   IDG,MCONV,ECONV,DCONV,LT,IAB
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMON /NUMDEN/ SNDO2(60), SNDN2(60), SNDO(60), SNDN(60),
     1                                 SNDE(60), SNDC(60),
     2                SNDH(60), SNDC2(60), SNDH2(60), SNDCO(60),
     3                SNDC3(60),SNDC2H(60)
      COMMON /SPEC/ MF, XMOL
C
      DO100I=1,20
      DO100J=1,NETA
  100 SY(I,J) = C(I,J)
      DO200I=1,20
      DO200J=1,NETA
  200 IF(C(I,J).LT.0.)C(I,J) = 1.E-20
C
C  **  NETA = NUMBER OF ETA POINTS
C      MF   = 1  IF SPECIE  MOLE FRACTIONS ARE INPUT AND NUMBER DENSITY
C                TO BE COMPUTED
C             0  IF SPECIE NUMBER DENSITIES ARE INPUT
```

```
C       NS    = NUMBER OF SPECIES TO BE INPUT
C       LINES= 1  IF LINE CALCULATION IS TO BE DONE
C              0  IF ONLY CONTINUUM CALCULATION IS TO BE DONE
C       IDG   = 0  ONLY FINAL PRINT IS GIVEN
C              1  PRINT IS GIVEN FOR EACH ETA
C             99  COMPLETE PRINT
C
C
C   **  R         = BODY RADIUS (FT)
C       DELTA     = NONDIMENSIONAL STAND-OFF DISTANCE
C       DTIL      = TRANSFORMED STAND-OFF DISTANCE
C       XMOL      = 1.0 FOR RUN WITH MOLECULES
C                   0.0 FOR RUN WITHOUT MOLECULES
C
      XMOL  = 0.0
      XMOL= 1.0
C
C     ** DETERMINE ETZ POINTS
      N2 = NETA-2
      K =0
      IEZ =0
      DO 20 I=1,N2,2
      K=K+1
 20   ETZ(K) = ETA(I)
      ETZ(K+1) = ETA(NETA-1)
      ETZ(K+2) = ETA(NETA)
      IEZ = K +1
C
C     **  COMPUTE THE  Y  COORDINATE   **
      YOND(1) = 0.0
      SUM = 0.0
      DO 30 K=2,NETA
      DETA = ETA(K) -ETA(K-1)
      SUM = SUM +DETA*(1./RHO(K) +1./RHO(K-1) )/2.0
      YOND(K) = DTIL *SUM
```

```
 30    CONTINUE
       DELTA = YOND(NETA)
       DO 40 K=1,NETA
       YOND(K) = YOND(K)/YOND(NETA)
 40    CONTINUE
C
       LINES= 1
       IDGS = IDG
       IDG = 0
       CALL TRANS(1)
       IDG= IDGS
C
C  **   INDEX IS NUMBER GIVEN SPECIE FOR USE IN STORING ARRAYS **
C                          1     =     O2
C                          2     =     N2
C                          3     =     O
C                          4     =     N
C                          5     =     E-
C                          6     =     C
C                          7     =     H
C                          8     =     C2
C                          9     =     H2
C                         10     =     CO
C                         11     =     C3
C                         12     =     C2H
C
C
       DO300I=1,20
       DO300J=1,NETA
   300 C(I,J) = SY(I,J)
       RETURN
       END
```

```
      SUBROUTINE RADIN
      COMMON /DBUG/ QLC(60), QCL(60), QLL(60), DQN(60), QCC(60),
     1              BEEC(12,60),   FMUC(12,60),   EM(12,60),
     2              EP(12,60),     TAUC(12,60),   BEEL(9,60),
     3              QCCP(12),      WMM(9,60),     GMM(9,60),
     4              EEM(9,60),     XLMM(9,60),    QLCF(9),
     5              QCLP(9),       QLLP(9),       DELTA,    IY,   IYY,
     6              WPP(9,60),     GPP(9,60),     EEP(9,60),
     7              XLPP(9,60),    FG(9,4),       GP(9,4),
     8              WN(9,4),       FMUL(9,60),    SSM(9,4,60),
     9              GGM(9,4,60),   ETAM(9,4,60),  SBM(9,4,60),
     A              TAUL(9,60)
C  **  GROUP 1  **
      WN(1,1)=0.
      FG(1,1)=0.
      GP(1,1)=0.
      WN(1,2)=18.
      WN(1,3)=15.
      WN(1,4)=5.
C  **  GROUP 2  **
      WN(2,1)=3.0
      WN(2,2)=5.0
      WN(2,3)=11.0
      WN(2,4)=10.
C  **  GROUP 3  **
      WN(3,1)=0.
      FG(3,1)=0.
      GP(3,1)=0.
      WN(3,2)=2.0
      WN(3,3)=0.
      FG(3,3)=0.
      GP(3,3)=0.
      WN(3,4)=0.
      FG(3,4)=0.
      GP(3,4)=0.
```

```
C   **   GROUP 4   **
      WN(4,1)=0.
      FG(4,1)=0.
      GP(4,1)=0.
      WN(4,2)=8.0
      WN(4,3)=2.0
      WN(4,4)=0.
      FG(4,4)=0.
      GP(4,4)=0.
C   **   GROUP 5   **
      WN(5,1)=0.
      FG(5,1)=0.
      GP(5,1)=0.
      WN(5,2)=14.
      WN(5,3)=4.0
      WN(5,4)=1.0
C   **   GROUP 6   **
      WN(6,1)=1.0
      WN(6,2)=4.0
      WN(6,3)=13.0
      WN(6,4)=2.0
C   **   GROUP 7   **
      WN(7,1)=0.
      FG(7,1)=0.
      GP(7,1)=0.
      WN(7,2)=6.0
      WN(7,3)=14.0
      WN(7,4)=3.0
C   **   GROUP 8   **
      WN(8,1)=2.0
      WN(8,2)=2.0
      WN(8,3)=11.
      WN(8,4)=15.
C   **   GROUP 9   **
      WN(9,1)=0.
```

```
      FG(9,1)=0.
      GP(9,1)=0.
      WN(9,2)=1.0
      WN(9,3)=11.
      WN(9,4)=10.
      RETURN
      END
```

```
      SUBROUTINE TRANS (ISW)
C
C-----THIS IS A MODIFIED VERSION OF SUBROUTINE TRANS FROM K WILSON
C     TRANS IS DOCUMMENTED IN  LMSC-687209 APRIL 69  ---------------
C
      COMMON /ZPI/ ZPO(6), ZPN(6),ZPH(2),  ZPC(7)
      COMMON /FINV/ NHVL,NIHVC,FHVC(12),DJ(9),HVJ(9),ZKZ
      COMMON /SFLUX/ QRI(3)
      COMMON /TRN/        NUT(60), FMC(12,60), FPC(12,60),
     1            FM(9,60), FP(9,60), LINES
      COMMON /YL/ETA(60),  YD(60)
      COMMON/PROPI/P (60),  R(60), T(60),AMW(60),C (20,60),EC(5,60)
      COMMON /FRSTRM/ U INF, RINF, UINF2,XL , RE, LXI, ITM, IEM, NES
      COMMON /DEL/  W(1),DTIL,DTILS
      COMMON /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CFNF
      COMMON /MAIM/KEEP,MAXE,MAXM,MAXD,    IDG,MCONV,ECONV,DCONV,LT,IAB
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMON /RH/ DUD,DPHI,TD,RZB,PD,HD,HTOTAL
      COMMON/WT/SMW(20),AWT(5)
      COMMON /TEST/ETZ(60),IEZ
      COMMON /NUMDEN/ SNDO2(60), SNDN2(60), SNDO(60), SNDN(60),
     1                                       SNDE(60), SNDC(60),
     2              SNDH(60), SNDC2(60), SNDH2(60), SNDCO(60),
     3              SNDC3(60),SNDC2H(60)
      COMMON /DBUG/ QLC(60), QCL(60), QLL(60), DQN(60), QCC(60),
     1            BEEC(12,60),    FMUC(12,60),    EM(12,60),
     2            EP(12,60),      TAUC(12,60),    BEEL(9,60),
     3            QCCP(12),       WMM(9,60),      GMM(9,60),
     4            EEM(9,60),      XLMM(9,60),     QLCP(9),
     5            QCLP(9),        QLLP(9),        DELTA,     IY,  IYY,
     6            WPP(9,60),      GPP(9,60),      EEP(9,60),
     7            XLPP(9,60),     FG(9,4),        GP(9,4),
     8            WN(9,4),        FMUL(9,60),     SSM(9,4,60),
     9            GGM(9,4,60),    ETAM(9,4,60),   SBM(9,4,60),
     A            TAUL(9,60)
```

```
      COMMON /SPEC/ MF, XMOL
      DIMENSION  XKT(60),          DQ(60)
C
C **   BAND AVERAGE ABSORPTION CROSS SECTION (EQ.A2)  **
C
      SIGMA(ZH,ZA,ZB,ZG)=  ((5.0E+03*T1*ZG*ZKZ)/BE) * (EXP(ZDL/T1)
     1                     *ZH*(ZA+ZB* (ZH**2)/3.0) +
     2                     T1 * (ZA+2.0*ZB*T12) -T1*EXP((ZH-ZHVP)/T1)
     3                     *(ZA+ZB*(ZHVP-ZH)**2) -T1*EXP((ZH-ZHVP)/T1)
     4                     *2.0*ZB*T1*(ZHVP-ZH+T1))
      SIGMA2(ZH,ZG,ZE,ZY)=7.26E-16*T1*ZG*EXP((-ZE+ZY+ZDL)/T1)/ZH**3
      GAMMA(ZX)=(1.0+(1.5707963*ZX)**1.25)**(-0.4)
      XLAMB(ZX)=(1.0+ZX*EXP(-ZX))/SQRT(1.0+6.283185 *ZX)
C
C **   W(GROUP)/D CORRELATION (EQ.88)  **
C
      PHI1(ZX)=(ATAN(1.570796 *ZX)/1.570796 )
C
C **   FLUX DIVERGENCE OVERLAPPING FUNCTION (EQ.92)  **
C
      PHI2(ZX)=EXP(-ZX)
C
      DO 400 I=1,NES
  400 T(I)= T(I)*TD
      ZHVP=5.0
      YI=0.0
      CONVER = 3.10375E+23 *R  (I) *RDZ
      SNDE(NES) = CONVER * C( 7,NES)/SMW(7)
      XNE=SNDE(NES)
      FNE=(4.71E-6 * XNE**(2.0/7.0))/((T(NES)/11606.)**(1.0/7.0))
      ZDL=AMIN1(0.20,FNE)
C
C **   DEBUG PRINT  **
C
      IF (IDG.NE.0) CALL BUGPR (1)
```

```
      DELTA=W(1) * XL * 30.48006
      CALL BUGPR (2)
 6001 CONTINUE
      DO 91 L=1,NES
      XKT(L)=T(L)/11606.
      T1=XKT(L)
      CALL SND(L)
C
C **   PARTITION FUNCTIONS FOR H, C, N, O   **
C
   94 IF(T(L).GT.15000.) GO TO 6
C
C **   LOW  TEMPERATURE   **
C
      SUMH=2.0
      SUMC=9.0  +  5.0 * EXP(-1.264/T1) + EXP(-2.684/T1) +
     1              5.0 * EXP(-4.183/T1)
      SUMN=4.0 + 10.0 *EXP(-2.384/T1) + 6.0 *EXP(-3.576/T1)
      SUMO= 9.0 + 5.0 * EXP(-1.975/T1)
      GO TO 7
C
C **   HIGH TEMPERATURE   **
C
    6 SUMH=2.0
      SUMC=2.71818  +  6.40677 * T(L)/1.0E4 -0.45466 * (T(L)/1.0E4)**2
      SUMN=5.938216 - 0.225593 * T(L)/1.0E3 + 0.015408 * (T(L)/1.0E3)**2
      SUMO=11.79563 -0.317964 * T(L)/1.0E3 + 0.013765 * (T(L)/1.0E3)**2
    7 CONTINUE
      T12=T1**2
      GH = 6.4994
      DO 5 K=1,12
      GF=FHVC(K)/T1
      GHM=GH
      GH=EXP(-GF)  *GF  *  (GF**2 + 3.0 *GF  +6.0  + 6.0/GF)
C
```

```
C **    PLANK MEAN ABSORPTION COEFFICIENT FOR BAND INTERVALS (EQ.A3)  ** TRAN1060
C                                                                       TRAN1070
      BEEC(K,L)=5.04E3  *  (T12**2)  *  (GHM-GH)                        TRAN1080
      BE=BEEC(K,L)                                                      TRAN1090
C                                                                       TRAN1100
C **    ABSORPTION CROSS SECTICNS  **                                   TRAN1110
C       SOECIES --                                                      TRAN1120
C                   N       N2       CO                                 TRAN1130
C                   O       O2                                          TRAN1140
C                   C       C2       C2H                                TRAN1150
C                   H       H2       C3                                 TRAN1160
C                                                                       TRAN1170
      SGH=0.                                                            TRAN1180
      SGN=0.                                                            TRAN1190
      SGC=0.                                                            TRAN1200
      SGO=0.                                                            TRAN1210
      SGCO=0.                                                           TRAN1220
      SGC2=0.                                                           TRAN1230
      SGO2=0.                                                           TRAN1240
      SGN2=0.                                                           TRAN1250
      SGH2=0.                                                           TRAN1260
      SGC3=0.                                                           TRAN1270
      SGC2H = 0.0                                                       TRAN1280
      GO TO (581,582,583,584,585,586,587,588,589,590,591,592),K         TRAN1290
  581 SGH=SIGMA(2.4,1.0,0.0,1.0)  *  EXP(-13.56/T1)                     TRAN1300
      SGC=SIGMA(3.78,  0.3,   0.0488,  1.33)  * EXP(-11.26/T1)          TRAN1310
      SGN=SIGMA(4.22,  0.24,  0.0426,  4.5) * EXP(-14.54/T1)            TRAN1320
      SGO=SIGMA(4.22,  0.24,  0.0426,   .8888889)  * EXP(-13.61/T1)     TRAN1330
      GO TO 38                                                          TRAN1340
  582 ZZHV=5.5                                                          TRAN1350
      SGC2=8.0E-18 * EXP(-0.5/T1)  +  3.0E-18                           TRAN1360
      SGC3=4.0E-18                                                      TRAN1370
  593 CALL ZHV(ZZHV,ZZO,ZZN,ZZI,ZZC)                                    TRAN1380
      SGC=SIGMA2(ZZHV, 1.33,  11.26,  3.78) * ZZC + SGC                 TRAN1390
      SGN=SIGMA2 (ZZHV,4.50,  14.54,  4.22) * ZZN                       TRAN1400
```

```
594 SGO=SIGMA2 (ZZHV, .889,  13.61,  4.22) * ZZO                       TRAN1410
595 SGH=SIGMA2 (ZZHV, 1.00,  13.56,  2.40)                             TRAN1420
    GO TO 38                                                           TRAN1430
583 ZZHV=6.5                                                           TRAN1440
    SGC2=1.0E-18                                                       TRAN1450
    SGCO=3.0F-18 * EXP(-0.7/T1)                                        TRAN1460
    GO TO 593                                                          TRAN1470
584 ZZHV=7.5                                                           TRAN1480
    SGC=5.0E-17  * EXP(-4.18/T1)/SUMC                                  TRAN1490
    SGCO=1.9E-17  * EXP(-0.5/T1)                                       TRAN1500
    SGO2=6.0E-19                                                       TRAN1510
    SGC2H = 1.3E-18                                                    TRAN1520
    GO TO 593                                                          TRAN1530
585 ZZHV=8.5                                                           TRAN1540
    SGC=5.0E-17 *EXP(-4.18/T1)/SUMC  +                                 TRAN1550
   1     2.2E-17* EXP(-2.68/T1)/SUMC                                   TRAN1560
    SGCO=2.5E-17                                                       TRAN1570
    SGO2=2.0E-19                                                       TRAN1580
    SGC2H = 8.5E-19                                                    TRAN1590
    GO TO 593                                                          TRAN1600
586 ZZHV=9.5                                                           TRAN1610
    SGC=5.0E-17 * EXP(-4.18/T1)/SUMC  +                                TRAN1620
   1     2.2E-17 * EXP(-2.68/T1)/SUMC                                  TRAN1630
    SGCO=5.0E-18                                                       TRAN1640
    SGO2=1.0E-18                                                       TRAN1650
    GO TO 593                                                          TRAN1660
587 SGN=3.2E-18 *T1 *EXP(-10.2/T1)/SUMN                                TRAN1670
    SGO2=6.0E-19                                                       TRAN1680
    ZZHV=10.4                                                          TRAN1690
    CALL ZHV(ZZHV,ZZO,ZZN,ZZI,ZZC)                                     TRAN1700
596 SGC=(8.5E-17 *EXP(-1.26/T1)  +  2.2E-17 * EXP(-2.75/T1)            TRAN1710
   1     +  5.0E-17 * EXP(-4.18/T1))/SUMC                              TRAN1720
    GO TO 594                                                          TRAN1730
588 ZZHV=10.9                                                          TRAN1740
    CALL ZHV(ZZHV,ZZO,ZZN,ZZI,ZZC)                                     TRAN1750
```

```
      SGN=(5.16E-17  *EXP(-3.50/T1))/SUMN
      GO TO 596
  589 ZZHV=11.6
      CALL ZHV(ZZHV,ZZO,ZZN,ZZI,ZZC)
      SGN2=1.0F-18
      SGN=(5.16E-17 *  EXP(-3.50))/SUMN
  598 SGC=(9.9E-17  +  8.5E-17 * EXP(-1.26/T1)  +2.2E-17 * EXP(-2.75/T1)
     1      +  5.0E-17 * EXP(-4.18/T1))/SUMC
      IF (K.LT.11) GO TO 594
      GO TO 38
  590 ZZHV=12.7
      CALL ZHV  (ZZHV,ZZO,ZZN,ZZI,ZZC)
      SGN2=2.0F-18
      SGH2 = 2.7E-17
  599 SGN=(6.4E-17 * EXP(-2.30/T1)  +  5.16E-17 * EXP(-3.50/T1))/SUMN
     1      +  SGN
      GO TO 598
  591 SGH=1.18F-17/SUMH
      SGO=3.6E-17/SUMO
      SGN2=1.0E-17
      SGH2 = 2.7E-17
      GO TO 599
  592 SGN=3.6E-17/SUMN
      SGN2=1.0E-18
      GO TO 599
   38 CONTINUE
      FMUC(K,L)= SNDH(L)*SGH + SNDC(L)*SGC + SNDN(L)*SGN + SNDO(L)*SGO
     1           + XMOL *     (SNDN2(L)*SGN2 + SNDO2(L)*SGO2 +
     2           SNDC2(L)*SGC2 + SNDH2(L)*SGH2 + SNDCO(L)*SGCO +
     3           SNDC3(L)*SGC3 +SNDC2H(L)*SGC2H )
      IF (L.GT.1) GO TO 8
      TAUC(K,L)=0.
      GO TO 5
    8 TAUC(K,L)=TAUC(K,L-1)+(YD(L)-YD(L-1))*
     1      (FMUC(K,L-1)+FMUC(K,L)) * DELTA
```

C-H

```
      5 CONTINUE
        IF (LINES.EQ.0) GO TO 91
C
C **    FRACTIONAL POPULATION STATES FOR H, N, O, C  **
C
        CALL ZP (T1,SUMN,SUMO,SUMH,SUMC)
C **    CALCULATION OF PARAMETERS FOR 9 LINE GROUPES  **
C       WN -- NUMBER OF LINES
C       FG -- EFFECTIVE F-NUMBER
C       GP -- EFFECTIVE HALF-WIDTH
C
C GROUP 1
        FG(1,2)=(1.02 * ZPC(5)  +  .795 * ZPC(6)  +  0.114 * ZPC(7))
      1    /WN(1,2)
        GP(1,2)=(8.16E-11 * SQRT(ZPC(5))  +  1.25E-10 * SQRT(ZPC(6))
      1      +2.55F-10 * SQRT(ZPC(7)))**2 /(FG(1,2)* WN(1,2)**2)
        FG(1,3)=(1.040 * ZPN(4)  +  1.29 * ZFN(5)  +  0.00 *ZPN(6))
      1          /WN(1,3)
        GP(1,3)=(6.65E-11 * SQRT(ZPN(4))  +  1.71E-10 * SQRT(ZPN(5))
      1         +  0.00E-10 * SQRT(ZPN(6)))**2/(FG(1,3) * WN(1,3)**2)
        FG(1,4)=(1.00 * ZPO(5)  +  .978 * ZPO(6))/WN(1,4)
        GP(1,4)=(3.90E-11 * SQRT(ZPO(5))  +  9.68E-11 * SQRT(ZPO(6)))**2
      1          /(FG(1,4) * WN(1,4)**2)
        FMUL(1,L)=FMUC(1,L)
C GROUP 2
        FG(2,1)=0.805 * ZPH(2)/WN(2,1)
        GP(2,1)=2.37E-10 * 2.37E-10 * ZPH(2)/(FG(2,1) * WN(2,1)**2)
        FG(2,2)=(0.00E-2 * ZPC(5)  +   6.71E-2 * ZPC(6))/WN(2,2)
        GP(2,2)=(0.00E-12 * SQRT(ZPC(5))  +  7.15E-11 * SQRT(ZPC(6)))**2
      1          /(FG(2,2) * WN(2,2)**2)
        FG(2,3)=(0.047* ZPN(4)  +  2.85E-2 * ZPN(5))/WN(2,3)
        GP(2,3)=(1.11E-10 * SQRT(ZPN(4))  +  6.07E-11 * SQRT(ZPN(5)))**2
      1          /(FG(2,3) * WN(2,3)**2)
        FG(2,4)=(.0217 * ZPO(4)  +  8.25E-2 * ZPO(5))/WN(2,4)
        GP(2,4)=(2.61E-11 * SQRT(ZPO(4))  +  7.19E-11 * SQRT(ZPO(5)))**2
```

```
     1   /(FG(2,4) * WN(2,4)**2)                                        TRAN2460
      FMUL(2,L)=FMUC(1,L)                                               TRAN2470
C   GROUP 3                                                             TRAN2480
      FG(3,2)=(7.29E-2 * ZPC(2)  +  6.76E-2 *ZPC(3))/WN(3,2)            TRAN2490
      GP(3,2)=(9.08E-12 * SQRT(ZPC(2))  +  8.75E-12 * SQRT(ZPC(3)))**2  TRAN2500
     1        /(FG(3,2) * WN(3,2)**2)                                   TRAN2510
      FMUL(3,L)=FMUC(2,L)                                               TRAN2520
C   GROUP 4                                                             TRAN2530
      FG(4,2)=(1.05 * ZPC(1)  +  1.10E-2 *ZPC(2)  +  0.150 * ZPC(3))    TRAN2540
     1        /WN(4,2)                                                  TRAN2550
      GP(4,2)=(9.57E-12 * SQRT(ZPC(1))  +  4.86E-12 * SQRT(ZPC(2))      TRAN2560
     1        +  5.93E-10 * SQRT(ZPC(3)))**2/(FG(4,2) * WN(4,2)**2)     TRAN2570
      FG(4,3)=( 7.40E-2  * ZPN(2)  +  6.34E-2 * ZPN(3))/WN(4,3)         TRAN2580
      GP(4,3)=(8.22E-12 * SQRT(ZPN(2))  +  7.60E-12 * SQRT(ZPN(3)))**2  TRAN2590
     1        /(FG(4,3) * WN(4,3)**2)                                   TRAN2600
      FMUL(4,L)=FMUC(4,L)                                               TRAN2610
C   GROUP 5                                                             TRAN2620
      FG(5,2)=(0.329 * ZPC(1)  +  0.118 * ZPC(2)  +  0.226 * ZPC(4))    TRAN2630
     1        /WN(5,2)                                                  TRAN2640
      GP(5,2)=(3.65E-11 * SQRT(ZPC(1))  +  5.77E-10 * SQRT(ZPC(2))      TRAN2650
     1        +  6.56E-11 *  SQRT(ZPC(4)))**2/(FG(5,2) * WN(5,2)**2)    TRAN2660
      FG(5,3)=0.108 * ZPN(3)/WN(5,3)                                    TRAN2670
      GP(5,3)=3.09E-11 * 3.09E-11 * ZPN(3)/(FG(5,3) * WN(5,3)**2)       TRAN2680
      FG(5,4)=4.71E-2 * ZPO(1)/WN(5,4)                                  TRAN2690
      GP(5,4)=5.08E-12 * 5.08E-12 * ZPO(1)/(FG(5,4) * WN(5,4)**2)       TRAN2700
      FMUL(5,L)=FMUC(6,L)                                               TRAN2710
C   GROUP 6                                                             TRAN2720
      FG(6,1)=0.416 * ZPH(1)/WN(6,1)                                    TRAN2730
      GP(6,1)=3.02E-11 * 3.02E-11 * ZPH(1)/(FG(6,1)* WN(6,1)**2)        TRAN2740
      FG(6,2)=8.65E-2 * ZPC(1)/WN(6,2)                                  TRAN2750
      GP(6,2)=2.35E-10 * 2.35E-10 * ZPC(1)/(FG(6,2) * WN(6,2)**2)       TRAN2760
      FG(6,3)=(0.184 * ZPN(1)  +  0.290 * ZPN(2)  +  8.52E-2 * ZPN(3))  TRAN2770
     1        /WN(6,3)                                                  TRAN2780
      GP(6,3)=(1.07E-11 * SQRT (ZPN(1))  +  4.28E-11 * SQRT(ZPN(2))     TRAN2790
     1        +  2.09E-10 * SQRT(ZPN(3)))**2/(FG(6,3) * WN(6,3)**2)     TRAN2800
```

```
      FG(6,4)=(.120 * ZPO(2)  +  0.151 * ZPO(3))/WN(6,4)                TRAN2810
      GP(6,4)=(8.85E-12 * SQRT(ZPO(2))  +  9.93E-12 * SQRT(ZPO(3)))**2  TRAN2820
     1        /(FG(6,4) * WN(6,4)**2)                                   TRAN2830
      FMUL(6,L)=FMUC(7,L)                                               TRAN2840
C    GROUP  7                                                           TRAN2850
      FG(7,2)=(4.51E-2 * ZPC(1)  +  0.705 * ZPC(2))/WN(7,2)             TRAN2860
      GP(7,2)=(6.07E-10 * SQRT(ZPC(1))  +  2.10E-10 * SQRT(ZPC(2)))**2  TRAN2870
     1        /(FG(7,2) * WN(7,2)**2)                                   TRAN2880
      FG(7,3)=(0.454 * ZPN(1)  +  9.66E-2 * ZPN(2)                      TRAN2890
     1        +  0.178 * ZPN(3))/WN(7,3)                                TRAN2900
      GP(7,3)=(2.71E-12 * SQRT(ZPN(1))  +  2.34E-10 * SQRT(ZPN(2))      TRAN2910
     1        +  2.46E-11 * SQRT(ZPN(3)))**2/(FG(7,3)* WN(7,3)**2)      TRAN2920
      FG(7,4)=4.23E-2 * ZPO(3)/WN(7,4)                                  TRAN2930
      GP(7,4)=2.52E-11 * 2.52E-11 * ZPO(3)/(FG(7,4) * WN(7,4)**2)       TRAN2940
      FMUL(7,L)=FMUC(9,L)                                               TRAN2950
C   GROUP  8                                                            TRAN2960
      FG(8,1)=0.108 * ZPH(1)/WN(8,1)                                    TRAN2970
      GP(8,1)=1.32E-10  *  1.32E-10 * ZPH(1)                            TRAN2980
     1        /(FG(8,1) * WN(8,1)**2)                                   TRAN2990
      FG(8,2)=(0.379 * ZPC(1)  +  1.05 * ZPC(3))/WN(8,2)                TRAN3000
      GP(8,2)=(1.95E-11 * SQRT(ZPC(1))  +  1.27E-10 * SQRT(ZPC(3)))**2  TRAN3010
     1        /(FG(8,2) * WN(8,2)**2)                                   TRAN3020
      FG(8,3)=(0.155 * ZPN(1)  +  0.142*ZPN(2)  +  3.75E-2 * ZPN(3))    TRAN3030
     1        /WN(8,3)                                                  TRAN3040
      GP(8,3)=(2.98E-11 * SQRT(ZPN(1))  +  7.08E-11 * SQRT(ZPN(2))      TRAN3050
     1        +  1.33E-10 * SQRT(ZPN(3)))**2/(FG(8,3) * WN(8,3)**2)     TRAN3060
      FG(8,4)=(0.146 * ZPO(1)  +  8.61E-2*ZPO(2)                        TRAN3070
     1        +  9.33E-2 * ZPO(3))/WN(8,4)                              TRAN3080
      GP(8,4)=(1.97E-10 * SQRT(ZPO(1))  +  1.80E-11 * SQRT(ZPO(2))      TRAN3090
     1        +  8.13E-11 * SQRT(ZPO(3)))**2/(FG(8,4) * WN(8,4)**2)     TRAN3100
      FMUL(8,L)=FMUC(10,L)                                              TRAN3110
C   GROUP  9                                                            TRAN3120
      FG(9,2)=2.95 * ZPC(2)/WN(9,2)                                     TRAN3130
      GP(9,2)=5.85E-12 * 5.85E-12 * ZPC(2)/(FG(9,2) * WN(9,2)**2)       TRAN3140
      FG(9,3)=(0.224 * ZPN(1)  +  2.92E-2 * ZPN(2))/WN(9,3)             TRAN3150
```

```
      GP(9,3)=(3.41E-10 * SQRT(ZPN(1))  +  1.48E-10 * SQRT(ZPN(2)))**2
     1        /(FG(9,3) * WN(9,3)**2)
      FG(9,4)=(5.24E-2 * ZPO(1)  +  7.22E-2 * ZPO(2)
     1        +  6.04E-2 * ZPO(3))/WN(9,4)
      GP(9,4)=(5.76E-12 * SQRT(ZPO(1))  +  7.20E-11 * SQRT(ZPO(2))
     1        +  8.05E-11 * SQRT(ZPO(3)))**2/(FG(9,4) * WN(9,4)**2)
      FMUL(9,L)=FMUC(11,L)
C
C **   PLANCK FUNCTION  **
C
      DO 9 J=1,NHVL
      BEEL(J,L)=5.04E3 * HVJ(J)**3 / (EXP(HVJ(J)/T1)  -  1.0)
C
C **   INDUCED EMISSICN FACTOR (EQ 81)  **
C
      SSM(J,1,L)=1.10E-16*SNDH (L)*(1.0-EXP(-HVJ(J)/T1)) * FG(J,1)
      SSM(J,2,L)=1.10E-16*SNDC (L)*(1.0-EXP(-HVJ(J)/T1)) * FG(J,2)
      SSM(J,3,L)=1.10E-16*SNDN (L)*(1.0-EXP(-HVJ(J)/T1)) * FG(J,3)
      SSM(J,4,L)=1.10E-16*SNDO (L)*(1.0-EXP(-HVJ(J)/T1)) * FG(J,4)
      DO 10 M=1,4
      GGM(J,M,L)=GP(J,M) * SNDE(L)   * (T(L)/1.0E4)**0.25
     1        +  1.0E-6
      IF(L.GT.1) GO TO 11
      ETAM(J,M,1)=0.
      SBM (J,M,1)=0.
      GO TO 10
   11 ETAM(J,M,L)=ETAM(J,M,L-1)+  (YD(L)-YD(L-1))
     1        *(SSM(J,M,L-1) * GGM(J,M,L-1)  +  SSM(J,M,L) * GGM(J,M,L))
     2        * DELTA/3.14159265
      SBM(J,M,L)=SBM(J,M,L-1)  +  (YD(L)-YD(L-1))
     1        * (SSM(J,M,L-1)+SSM(J,M,L)) * DELTA
   10 CONTINUE
      IF (L.GT.1) GO TO 12
      TAUL(J,1)=0.
      GO TO 9
```

```
   12 TAUL(J,L)=TAUL(J,L-1)  +  (YD(L)-YD(L-1))
     1          * (FMUL(J,L-1)+FMUL(J,L))  * DELTA
    9 CONTINUE
      IF (IDG.NE.99) GO TO 91
      CALL BUGPR (7)
C
   91 CONTINUE
      IEZ=IEZ+1
      ETZ(IEZ)=1.0
C
C **   CONTINUUM - CONTINUUM FLUX DIVERGENCE CALCULATION  **
C
      DO 300 K=1,IEZ
      DO 31 LK=1,NES
      I=LK
      NUT(K)=I
      IF (ABS(ETZ(K)-ETA(LK)) - 1.0E-5) 300,300,31
   31 CONTINUE
  300 CONTINUE
      DO 1612 J=1,9
      QCLP(J)=0.
      QLCP(J)=0.
      QLLP(J)=0.
      DO 1612 L=1,NES
      FM(J,L)=0.
 1612 FP(J,L)=0.
      DO 1613 L=1,IEZ
      QCL(L)=0.
      QLC(L)=0.
 1613 QLL(L)=0.
      DO 49 IYY=1,IEZ
      IY=NUT(IYY)
      DO 20 K=1,12
      FMC(K,IY)=0.
      FPC(K,IY)=0.
```

```
      IF (IY.EQ.1) GO TO 44                                             TRAN3860
      DO 40 L=1,IY                                                      TRAN3870
C                                                                       TRAN3880
C **   MINUS EMISSIVITY FUNCTION (EQ 47)   *                            TRAN3890
C                                                                       TRAN3900
      EM(K,L)=1.0  -  EXP(TAUC(K,L)-TAUC(K,IY))                         TRAN3910
      IF (L.EQ.1) GO TO 40                                              TRAN3920
C                                                                       TRAN3930
C **   MINUS CONTINUUM FLUX (EQ 46)   **                                TRAN3940
C                                                                       TRAN3950
      FMC(K,IY)=FMC(K,IY)  -  (EM(K,L)-EM(K,L-1))                       TRAN3960
     1         * (BEEC(K,L-1)+BEEC(K,L))/2.                             TRAN3970
   40 CONTINUE                                                          TRAN3980
   44 IF (IY.EQ.NES ) GO TO 41                                          TRAN3990
      DO 42 L=IY,NES                                                    TRAN4000
C                                                                       TRAN4010
C **   POSITIVE EMISSIVITY FUNCTION (EQ 47)   **                        TRAN4020
C                                                                       TRAN4030
      EP(K,L)=1.0  -  EXP(TAUC(K,IY)-TAUC(K,L))                         TRAN4040
      IF (L.EQ.IY) GO TO 42                                             TRAN4050
C                                                                       TRAN4060
C **   POSITIVE EMISSIVITY CONTINUUM FLUX (EQ 46)   **                  TRAN4070
C                                                                       TRAN4080
      FPC(K,IY)=FPC(K,IY)  +  (EP(K,L)-EP(K,L-1))                       TRAN4090
     1         * (BEEC(K,L-1)+BEEC(K,L))/2.                             TRAN4100
   42 CONTINUE                                                          TRAN4110
C                                                                       TRAN4120
C **   POSITIVE EMISSIVITY CONTINUUM FLUXDIVERGENCE (EQ 51)   **        TRAN4130
C                                                                       TRAN4140
   41 QCCP(K)=6.2831853 * FMUC(K,IY) *                                  TRAN4150
     1         (FMC(K,IY)  +  FPC(K,IY)  -  2.0* BEEC(K,IY))            TRAN4160
      FMC(K,IY)=FMC(K,IY) * 3.14159265                                  TRAN4170
      FPC(K,IY)=FPC(K,IY) * 3.14159265                                  TRAN4180
   20 CONTINUE                                                          TRAN4190
C                                                                       TRAN4200
```

```
C **    DEBUG PRINT   **
C
      IF (IDG.NE.99) GO TO 21
      CALL BUGPR (3)
   21 QCC(IYY)=0.
      DO 24 K=1,12
C
C **    LINE AND CROSS TERM FLUX DIVERGENCE CALCULATION  **
C
   24 QCC(IYY)=QCC(IYY)  +  QCCP(K)
      IF (LINES.EQ.0) GO TO 1614
C
C **    INTEGRATION FROM 1 TO IY   **
C
      IF (IY.EQ.1) GO TO 68
      DO 65 J=1,9
      DO 66 L=1,IY
      WIM=0.
      SUM1=0.
      SUM2=0.
      DO 67 M=1,4
      DIF=ETAM(J,M,IY)  -  ETAM(J,M,L)
      DIFSBM =      SBM(J,M,IY)-SBM(J,M,L)
      IF(ABS(DIFSBM).LT.1.E-10)  DIFSBM = 1.E-10
      BETAM=DIF / ( DIFSBM                )  *  3.14159265
      IF (L.EQ.IY) BETAM=GGM(J,M,L)
      IF(ABS(DIF).GT.1.E-10) GO TO 9001
      TM = 1.E-10
      GO TO 9002
 9001 CONTINUE
      TM=DIF/2.0/BETAM**2
 9002 RRM=DIF/2.0/GGM(J,M,IY)**2
      WWM=6.2831853 * WN(J,M) * BETAM * GAMMA(TM) * TM
      SUM1=SUM1  +  GAMMA(TM) * WN(J,M) * SSM(J,M,IY)
      SUM2=SUM2  +  XLAMB(RRM) * WN(J,M) * SSM(J,M,IY)
```

```
   67 WIM=WIM  +  WWM
      ALPHAM=WIM/DJ(J)
C
C **   OVERLAPPING LINE CALCULATIONS  **
C
C
C **   GROUP EQUIVALENT WIDTHS  (EQ.88)  **
C
      WMM(J,L)=DJ(J) * PHI1(ALPHAM) * EXP(TAUL(J,L)-TAUL(J,IY))
C
C **   GROUP GAMMA -- LINE TRANSPORT FUNCTION (EQ.92)  **
C
      GMM(J,L)=PHI2(ALPHAM) * SUM1
C
C **   MINUS EMISSIVITY FUNCTION FOR LINES (EQ.47)  **
C
      EFM(J,L)=1.0  -  EXP(TAUL(J,L)-TAUL(J,IY))
   66 XLMM(J,L)=PHI2(ALPHAM) * SUM2
   65 CONTINUE
      IF (IDG.EQ.99) CALL BUGPR(1)
      IF (IDG.EQ.99) CALL BUGPR (4)
   68 IF (IY.EQ.NES) GO TO 72
C
C **   INTEGRATION FROM IY TO NES  **
C
      DO 69 J=1,9
      DO 70 L=IY,NES
      WIP=0.
      SUM1=0.
      SUM2=0.
      DO 71 M=1,4
      DIF=ETAM(J,M,L)  -  ETAM(J,M,IY)
      DIFSBM =     SBM(J,M,L)-SBM(J,M,IY)
      IF(ABS(DIFSBM).LT.1.E-10)  DIFSBM = 1.E-10
      BETAP=DIF / ( DIFSBM               )  *  3.14159265
```

```
      IF (L.EQ.IY) BETAP=GGM(J,M,L)
      IF(ABS(DIF).GT.1.E-10) GO TO 9003
      TP = 1.E-10
      GO TO 9004
 9003 CONTINUE
      TP=DIF/2.0/BETAP**2
 9004 RRP=DIF/2.0/GGM(J,M,IY)**2
      WWP=6.2831853 * WN(J,M) * BETAP * GAMMA(TP) * TP
      SUM1=SUM1  +  GAMMA(TP) * WN(J,M) * SSM(J,M,IY)
      SUM2=SUM2  +  XLAMB(RRP) * WN(J,M) * SSM(J,M,IY)
   71 WIP=WIP+WWP
      ALPHAP=WIP/DJ(J)
      WPP(J,L)=DJ(J) * PHI1(ALPHAP) * EXP(TAUL(J,IY)-TAUL(J,L))
      GPP(J,L)=PHI2(ALPHAP) * SUM1
C
C **   POSITIVE EMISSIVITY FUNCTION FOR LINES (EQ.47)  **
C
      EEP(J,L)=1.0  -  EXP(TAUL(J,IY)-TAUL(J,L))
   70 XLPP(J,L)=PHI2(ALPHAP) * SUM2
   69 CONTINUE
C
C **   DEBUG PRINT  **
      IF (IDG.EQ.99) CALL BUGPR (5)
C
   72 DO 80 J=1,9
      ASM1=0.
      ASM2=0.
      FM(J,IY)=0.
      IF (IY.EQ.1) GO TO 81
      DO 82 L=2,IY
      FM(J,IY)=FM(J,IY)  -  (WMM(J,L)-WMM(J,L-1))
     1       *(BEEL(J,L-1)+BEEL(J,L)) * 1.5707963
      IF (L.EQ.IY) GO TO 82
      ASM1=ASM1  -  (EEM(J,L)-EEM(J,L-1))
     1        * (BEEL(J,L-1) * XLMM(J,L-1) + BEEL(J,L) * XLMM(J,L))/2.
```

```
      ASM2=ASM2  -  (XLMM(J,L)-XLMM(J,L-1))
     1         * (BEEL(J,L-1) * EXP(TAUL(J,L-1)-TAUL(J,IY)) + BEEL(J,L)
     2         * EXP(TAUL(J,L)-TAUL(J,IY)))/2.0
   82 CONTINUE
   81 ASP1=0.
      ASP2=0.
      IYP=IY+1
      IF (IY.EQ.NES) GO TO 83
      DO 84 L=IYP,NES
      FP(J,IY)=FP(J,IY)  +  (WPP(J,L)-WPP(J,L-1))
     1         * (BEEL(J,L-1)+BEEL(J,L)) * 1.5707963
      IF (L.EQ.IYP) GO TC 84
      ASP1=ASP1  +  (EEP(J,L)-EEP(J,L-1))
     1         * (BEEL(J,L-1) * XLPP(J,L-1) + BEEL(J,L) * XLPP(J,L))/2.0
      ASP2=ASP2  +  (XLPP(J,L)-XLPP(J,L-1)) *
     1         (BEEL(J,L-1) * EXP(TAUL(J,IY)-TAUL(J,L-1))  +  BEEL(J,L)
     2         * EXP(TAUL(J,IY)-TAUL(J,L)))/2.0
   84 CONTINUE
   83 QLCP(J)=2.0 * FMUL(J,IY) * (FM(J,IY)+FP(J,IY))
      SUMS=1.0
      SUMT=0.
      DO 86 M=1,4
   86 SUMT=SUMT  +  SSM(J,M,IY) * WN(J,M)
      ATM1=0.
      IF (IY.NE.1) ATM1=(BEEL(J,IY-1)+BEEL(J,IY)) /2.0 * EEM(J,IY-1)
     1         * XLMM(J,IY-1)
      ATP1=0.
      IF(IY.NE.NES) ATP1=(BEEL(J,IY+1)+BEEL(J,IY))/2.0 * EEP(J,IY+1)
     1         * XLPP(J,IY+1)
      QCLP(J)=6.2831853  * SUMS * (ASM1+ASP1+ATM1+ATP1)
      IF (IY.EQ.1) ATM2=-BEEL(J,IY) * SUMT
      IF (IY.NE.1) ATM2=(BEEL(J,IY-1)-BEEL(J,IY))  *  GMM(J,IY-1)
     1         -  BEEL(J,IY-1) * XLMM(J,IY-1)
      IF (IY.EQ.NES) ATP2=-BEEL(J,IY) * SUMT
      IF (IY.NE.NES)ATP2=(BEEL(J,IY+1)-BEEL(J,IY))  * GPP(J,IY+1)
```

```
     1           -  BEEL(J,IY+1) * XLPP(J,IY+1)
      QLLP(J)=6.2831853 * SUMS*(-ASM2-ASP2+ATM2+ATP2)
   80 CONTINUE
      QCL(IYY)=0.
      QLC(IYY)=0.
      QLL(IYY)=0.
      DO 85 J=1,9
      QCL(IYY)=QCL(IYY)   +  QCLP(J)
      QLC(IYY)=QLC(IYY)   +  QLCP(J)
   85 QLL(IYY)=QLL(IYY)   +  QLLP(J)
 1614 CONTINUE
      DQN(IYY)=-(QCC(IYY)+QCL(IYY)+QLC(IYY)+QLL(IYY))
C
C **    DEBUG PRINT   **
C
      IF (IDG.EQ.0) GO TO 49
      CALL BUGPR(6)
   49 CONTINUE
      IEZ=IEZ-1
      DQ(1)=DQN(1)
      L=2
      DO 1 N=2,NES
      DO 2 I=2,IEZ
      NP=I
      IF (ETZ(I).GT.ETA(N)) GO TO 3
    2 CONTINUE
    3 NN=NP-1
      AA=0.0
      ZB=(DQN(NN)-DQN(NP)) / (ETZ(NN)-ETZ(NP))
      CC=DQN(NN)   -  ZB * ETZ(NN)
      DQ(N)=AA * ETA(N)**2   +   ZE * ETA(N)   +   CC
      GO TO 1
    4 DQ(N)=DQN(NN)
    1 CONTINUE
C
```

```
C **   NON-DIMENSIONALIZE E(I)   **
      DO 250 I=1,NES
      T(I) = T(I)/TD
  250 E(I) = ((DQ(I)*XL )/(RINF*UINF**3))*20866.0 *RZB
      RETURN
      END
```

```
      SUBROUTINE TRANS2
      COMMON /SFLUX/ QRI(3)
      COMMON /YL/ETA(60),YOND(60)
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NES
      COMMON /TRN/        NUT(60), FMC(12,60), FPC(12,60),
     1              FM(9,60), FP(9,60), LINES
      COMMON /FINV/ NHVL,NIHVC,FHVC(12),DJ(9),HVJ(9),ZKZ
      COMMON /TEST/ETZ(60),IEZ
      COMMON /NUMDEN/ SNDO2(60), SNDN2(60), SNDO(60), SNDN(60),
     1                                    SNDE(60), SNDC(60),
     2                 SNDH(60), SNDC2(60), SNDH2(60), SNDCO(60),
     3                 SNDC3(60),SNDC2H(60)
      COMMON /SPEC/ MF, XMOL
      DIMENSION ETOUT(3)
      NETA=NES
      ETOUT(1)=0.0
      ETOUT(2)=0.5
      ETOUT(3)=1.0
      NOUT=3
C
C  OUTPUT FLUX
C
      WRITE (6,600)
      WRITE (6,603) (ETA(I),SNDN2(I),SNDO2(I),SNDN(I),SNDO(I),
     1                        SNDE(I), SNDH(I),
     2                       SNDC(I),SNDC2(I),SNDH2(I),SNDCO(I),SNDC3(I),
     3                 SNDC2H(I),               I=1,NETA)
C **   CONTINUUM CONTRIBUTICN TO THE SPECTRAL FLUX **
C
      WRITE (6,4103)
      DO 8040 K=1,NOUT
      DO 8041 LK=1,NES
      NUT(K)=LK
      IF (ABS(ETOUT(K)-ETA(LK)) - 1.0E-05) 8040,8040,8041
 8041 CONTINUE
```

```
 8040 CONTINUE
      L1=NUT(1)
      L2=NUT(2)
      L3=NUT(3)
      WRITE (6,8037)(ETOUT(IL),IL=1,3)
      FM1=0.0
      FP1=0.0
      FM2=0.0
      FP2=0.0
      FM3=0.0
      FP3=0.0
      DO 4104 KL=1,NIHVC
      WRITE (6,8042) KL,   FHVC(KL),   FMC(KL,L1),   FPC(KL,L1),
     1              FMC(KL,L2),   FPC(KL,L2),   FMC(KL,L3),   FPC(KL,L3)
      FM1=FM1   +  FMC(KL,L1)
      FP1=FP1   +  FPC(KL,L1)
      FM2=FM2   +  FMC(KL,L2)
      FP2=FP2   +  FPC(KL,L2)
      FM3=FM3   +  FMC(KL,L3)
      FP3=FP3   +  FPC(KL,L3)
 4104 CONTINUE
      WRITE (6,8045) FM1,   FP1,   FM2,   FP2,   FM3,   FP3
      QRI(1)=FM1+FP1
      QRI(2)=FM2+FP2
      QRI(3)=FM3+FP3
C
C **   LINE      CONTRIBUTION TO THE SPECTRAL FLUX **
C
      IF (LINES.EQ.0) RETURN
      WRITE (6,8035)
      WRITE (6,8037) (ETOUT(IL),IL=1,3)
      FM1=0.0
      FP1=0.0
      FM2=0.0
      FP2=0.0
```

```
      FM3=0.0
      FP3=0.0
C
C **   TOTAL FLUX CALCULATION  **
C
      DO 8043 KL=1,NHVL
      WRITE (6,8042) KL,   HVJ(KL),   FM(KL,L1),   FP(KL,L1),
     1              FM(KL,L2),   FP(KL,L2),   FM(KL,L3),   FP(KL,L3)
      FM1=FM1  +  FM(KL,L1)
      FP1=FP1  +  FP(KL,L1)
      FM2=FM2  +  FM(KL,L2)
      FP2=FP2  +  FP(KL,L2)
      FM3=FM3  +  FM(KL,L3)
      FP3=FP3  +  FP(KL,L3)
 8043 CONTINUE
      WRITE (6,8045) FM1,   FP1,   FM2,   FP2,   FM3,   FP3
      QRI(1)=QRI(1)  +  FM1  +  FP1
      QRI(2)=QRI(2)  +  FM2  +  FP2
      QRI(3)=QRI(3)  +  FM3  +  FP3
C
  600 FORMAT (1H1,33HNUMBER DENSITIES  (PARTICLES/CM3) ///5X,3HETA, 8X,
     1         2HN2, 8X,2HO2, 8X,1HN, 8X,1HC, 8X, 2HE-,8X,
     2 1HH,8X,1HC, 8X,2HC2, 8X,2HH2, 8X,2HCO, 8X,2HC3,8X,3HC2H///)
  603 FORMAT (1P13E10.2)
 4103 FORMAT (44H1CONTINUUM CONTRIBUTION TO THE SPECTRAL FLUX)
 8035 FORMAT (39H0LINE CONTRIBUTION TO THE SPECTRAL FLUX)
 8037 FORMAT (/22X,5HETA =F7.3,13X,5HETA =F7.3,13X,5HETA =F7.3//3X,1HI,
     1 3X,3HHNU,8X,6HQMINUS,7X,5HQPLUS,8X,6HQMINUS,7X,5HQPLUS,8X,
     2 6HQMINUS,7X,5HQPLUS/)
 8042 FORMAT (I4,F8.3,1P6E13.3)
 8045 FORMAT (12H0TOTAL FLUX ,1P8E13.3)
      RETURN
      END
```

```
      SUBROUTINE SND(I)
      COMMON/PROP1/PI(60),RHO(60), T(60),AMW(60),C (20,60),EC(5,60)
      COMMON /RFLUX/ E(60),IRAD,ITYPE
      COMMON /NON/RDZ,MUDZ,RMDZ,AKNF,HNF,CFNF
      COMMON/WT/SMW(20),AWT(5)
      COMMON /NUMDEN/ SNDO2(60), SNDN2(60), SNDO(60), SNDN(60),
     1                                      SNDE(60), SNDC(60),
     2                SNDH(60), SNDC2(60), SNDH2(60), SNDCO(60),
     3                SNDC3(60),SNDC2H(60)
C     **  CALCULATE SPECIE NUMBER DENSITIES BASED ON MOLE FRACTIONS  **
C
      CONVER = 3.10375E+23 *RHO(I) *RDZ
C
      SNDO2(I) = CCNVER * C( 1,I)/SMW( 1)
      SNDN2(I) = CONVER * C( 2,I)/SMW( 2)
      SNDO (I) = CCNVER * C( 3,I)/SMW( 3)
      SNDN (I) = CCNVER * C( 4,I)/SMW( 4)
      SNDE (I) = CCNVER * C( 7,I)/SMW( 7)
      SNDC (I) = CONVER * C( 8,I)/SMW( 8)
      SNDH (I) = CONVER * C( 9,I)/SMW( 9)
      SNDH2(I) = CCNVER * C(10,I)/SMW(10)
      SNDCO(I) = CCNVER * C(11,I)/SMW(11)
      SNDC3(I) = CCNVER * C(12,I)/SMW(12)
      SNDC2(I) = CONVER * C(19,I)/SMW(19)
      SNDC2H(I)= CCNVER * C(14,I)/SMW(14)
      RETURN
      END
```

```
      SUBROUTINE ZP(T1,SUMN,SUMO,SUMH,SUMC)
C
C     **  FRACTIONAL POPULATION STATES FOR N, O, H, C  **
C
      COMMON /ZPI/ ZPO(6), ZPN(6),ZPH(2),  ZPC(7)
      ZPH(1)=2.0/SUMH
      ZPH(2)=8.0 * EXP(-10.20/T1)/SUMH
      ZPC(1)=9.0/SUMC
      ZPC(2)=5.0 * EXP(-1.264/T1)/SUMC
      ZPC(3)=EXP(-2.684/T1)/SUMC
      ZPC(4)=5.0 * EXP(-4.183/T1)/SUMC
      ZPC(5)=12.0 * EXP(-7.532/T1)/SUMC
      ZPC(6)=36.0*EXP(-8.722/T1)/SUMC
      ZPC(7)=60.0 * EXP(-9.724/T1)/SUMC
      ZPN(1)=4.0/SUMN
      ZPN(2)=10.0* EXP(-2.384/T1)/SUMN
      ZPN(3)=6.0 * EXP(-3.576/T1)/SUMN
      ZPN(4)=18.0 * EXP(-10.452/T1)/SUMN
      ZPN(5)=54.0 * EXP(-11.677/T1)/SUMN
      ZPN(6)=90.0 * EXP(-13.002/T1)/SUMN
      ZPO(1)=9.0/SUMO
      ZPO(2)=5.0 * EXP(-1.967/T1)/SUMO
      ZPO(3)=EXP(-4.188/T1)/SUMO
      ZPO(4)=8.0 * EXP(-9.283/T1)/SUMO
      ZPO(5)=24.0 * EXP(-10.830/T1)/SUMO
      ZPO(6)=40.0 * EXP(-12.077/T1)/SUMO
C
      RETURN
      END
```

```
      SUBROUTINE BUGPR (IDGSW)
      COMMON /FRSTRM/ U INF, RINF, UINF2, R , RE, LXI, ITM, IEM, NES
      COMMON /YL/ETA(60),  YD(60)
      COMMON /TRN/         NUT(60), FMC(12,60), FPC(12,60),
     1               FM(9,60),  FP(9,60), LINES
      COMMON /DBUG/ QLC(60), QCL(60), QLL(60), DQN(60), QCC(60),
     1               BEEC(12,60),   FMUC(12,60),   EM(12,60),
     2               EP(12,60),     TAUC(12,60),   BEEL(9,60),
     3               QCCP(12),      WMM(9,60),     GMM(9,60),
     4               EEM(9,60),     XLMM(9,60),    QLCP(9),
     5               QCLP(9),       QLLP(9),       DELTA,    IY,  IYY,
     6               WPP(9,60),     GPP(9,60),     EEP(9,60),
     7               XLPP(9,60),    FG(9,4),       GP(9,4),
     8               WN(9,4),       FMUL(9,60),    SSM(9,4,60),
     9               GGM(9,4,60),   ETAM(9,4,60),  SBM(9,4,60),
     A               TAUL(9,60)
      GO TO (10,20,30,40,50,60,70), IDGSW
   10                 WRITE (6,194)
  194 FORMAT (1H1)
      RETURN
   20 WRITE (6,7182) DELTA
 7182 FORMAT (7HODELTA=1PE14.7,3H CM)
      RETURN
   30 WRITE (6,190) IY,   YD(IY)
  190 FORMAT (4H1IY=I3,2X,3HYD=1PE12.5//2X,1HK,2X,1HL,7X,3HETA,13X,2HYD,
     1 13X,2HMU,11X,3HTAU,14X,1HE,11X,3HBEE//)
      DO 22 K=1,12
      IF (IY.EQ.1) GO TO 23
      WRITE(6,191) (K,   L,   ETA(L),   YD(L),   FMUC(K,L),   TAUC(K,L),
     1             EM(K,L),   BEEC(K,L),   L=1,IY)
  191 FORMAT (2I3,1P6E15.5)
      WRITE (6,192)
  192 FORMAT (//)
   23 IF (IY.EQ.NES) GO TO 22
      WRITE (6,191) (K,   L,   ETA(L),   YD(L),   FMUC(K,L),
```

```
   1              TAUC(K,L),   EP(K,L),   BEEC(K,L),   L=IY,NES)
  22 WRITE (6,193) FMC(K,IY),   FPC(K,IY),   QCCP(K)
 193 FORMAT (5H0FIM=1PE12.5 ,2X,4HFIP=E12.5,2X,5HQCCP=E12.5)
     RETURN
 40                 WRITE (6,195) IY,   YD(IY),   ((J,   L,   YD(L),
   1              WMM(J,L),   GMM(J,L),   XLMM(J,L),   EEM(J,L),
   2              BEEL(J,L),   L=1,IY),   J=1,9)
 195 FORMAT (4H0IY=I3,2X,3HYY=1PE12.5//2X,1HJ,2X,1HL,7X,2HYD,12X,3HWMM,
   1 12X,3HGMM,11X,4HXLMM,13X,3HEEM,13X,3HBEE//(2I3,6E16.5))
     RETURN
 50                 WRITE (6,196) IY,   YD(IY),   ((J,   L,   YD(L),
   1         WPP(J,L),   GPP(J,L),   XLPP(J,L),   EEP(J,L),
   2         BEEL(J,L),   L=IY,NES),   J=1,9)
 196 FORMAT (4H0IY=I3,2X,3HYY=1PE12.5//2X,1HJ,2X,1HL,7X,2HYD,13X,3HWPP,
   1 2X,3HGPP,11X,4HXLPP,13X,3HEEP,13X,3HBEE//(2I3,6E16.5))
     RETURN
 60  WRITE (6,198) IY,   ETA(IY),   YD(IY)
 198 FORMAT (4H0IY=I3,2X,4HETA=1PE12.5,2X,3HYY=E12.5//2X,1HJ,5X,3HQCC,
   1 11X,3HFMC,11X,3HFPC,11X,3HQCL,11X,3HQLC,11X,3HQLL,12X,2HFM,12X,
   2 2HFP,11X,3HDQN//)
     WRITE (6,199) (J,   QCCP(J),   FMC(J,IY),   FPC(J,IY),
   1              QCLP(J),   QLCP(J),   QLLP(J),   FM(J,IY),FP(J,IY),
   2              J=1,9)
 199 FORMAT (I3,1P8E14.5)
     WRITE (6,8069) (J,   QCCP(J),   FMC(J,IY),   FPC(J,IY),   J=10,12)
8069 FORMAT (I3,1P3E14.5)
     WRITE (6,200) QCC(IYY),   QCL(IYY),   QLC(IYY),   QLL(IYY),
   1              DQN(IYY)
 200 FORMAT (1H0,2X,1PE14.5,28X,3E14.5,28X,E14.5)
     RETURN
 70  WRITE (6,197) L,   ETA(L),   YD(L),   ((J,   M,   WN(J,M),
   1              FG(J,M),   GP(J,M),   FMUL(J,L),   TAUL(J,L),
   2              SSM(J,M,L),   GGM(J,M,L),   ETAM(J,M,L),   SBM(J,M,L),
   3              M=1,4),J=1,9)
 197 FORMAT (3H0L=I3,2X,4HETA=1PE12.5,2X,3HYD=E12.5//2X,1HJ,2X,1HM,7X,
```

```
     1 1HN,13X,1HF,13X,1HG,11X,3HFMU,11X,3HTAU,11X,3HSSM,11X,3HGGM,10X, BUGP 710
     2 4HETAM,11X,3HSBM//(2I3,9E14.5))                                   BUGP 720
       RETURN                                                            BUGP 730
       END                                                               BUGP 740
```

```
      SUBROUTINE ZHV(HV,ZO,ZN,ZI,ZC)
C
C
C  **  THIS SUBROUTINE CALCULATES THE QUANTUM MECHANICAL CORRECTION
C           FACTORS GIVEN A FREQUENCY (HV) **
C
      X= HV
      X2 =X*X
      X3 =X2*X
      X4 =X3*X
      X5 =X4*X
      X6 =X5*X
      X7 =X6*X
      IF (X -9.82) 1,1,2
    1 ZO = .9999795           -.3155480*X          +2.824548 E-02*X2
     1     +6.677328 E-03*X3  -3.644585 E-03*X4    +8.058070 E-04*X5
     2     -7.708637 E-05*X6  +2.668133 E-06*X7
      GO TO 3
    2 ZO = (X/9.82)**3
    3 IF (X -8.35) 4,4,5
    4 ZN = 1.000148           - .4183535 *X        + .1680359 *X2
     1     -9.779458 E-02*X3  +3.354635 E-02*X4    -5.609353 E-03*X5
     2     +4.515535E-04*X6   -1.403585 E-05*X7
      GO TO 6
    5 ZN = (X/8.35)**3
    6 Y = X/4.0
      IF (Y-6.6) 9,9,10
    9 Y2 =Y*Y
      Y3 =Y2*Y
      Y4 =Y3*Y
      Y5 =Y4*Y
      Y6 =Y5*Y
      Y7 =Y6*Y
      ZI = 1.000379           - .2964767 *Y        +7.505242 E-02*Y2
     1     -1.702948E-02*Y3   +3.279554 E-03*Y4    -2.128469 E-04*Y5
```

```
      GO TO 11                                                          ZHV( 360
   10 ZI = (Y/6.6)**3                                                   ZHV( 370
   11 IF (X-7.37) 12,12,13                                              ZHV( 380
   12 ZC =  .9974367          - .4341812 *X        +8.531314 E-02*X2    ZHV( 390
     1     -1.393917 E-02*X3  +4.038545 E-03*X4    -5.426425 E-04*X5    ZHV( 400
     2     +2.812126 E-05*X6  -3.883530 E-07*X7                         ZHV( 410
      GO TO 14                                                          ZHV( 420
   13 ZC = (X/7.37)**3                                                  ZHV( 430
   14 RETURN                                                            ZHV( 440
      END                                                               ZHV( 450
```

APPENDIX C

DETAILED RESULTS

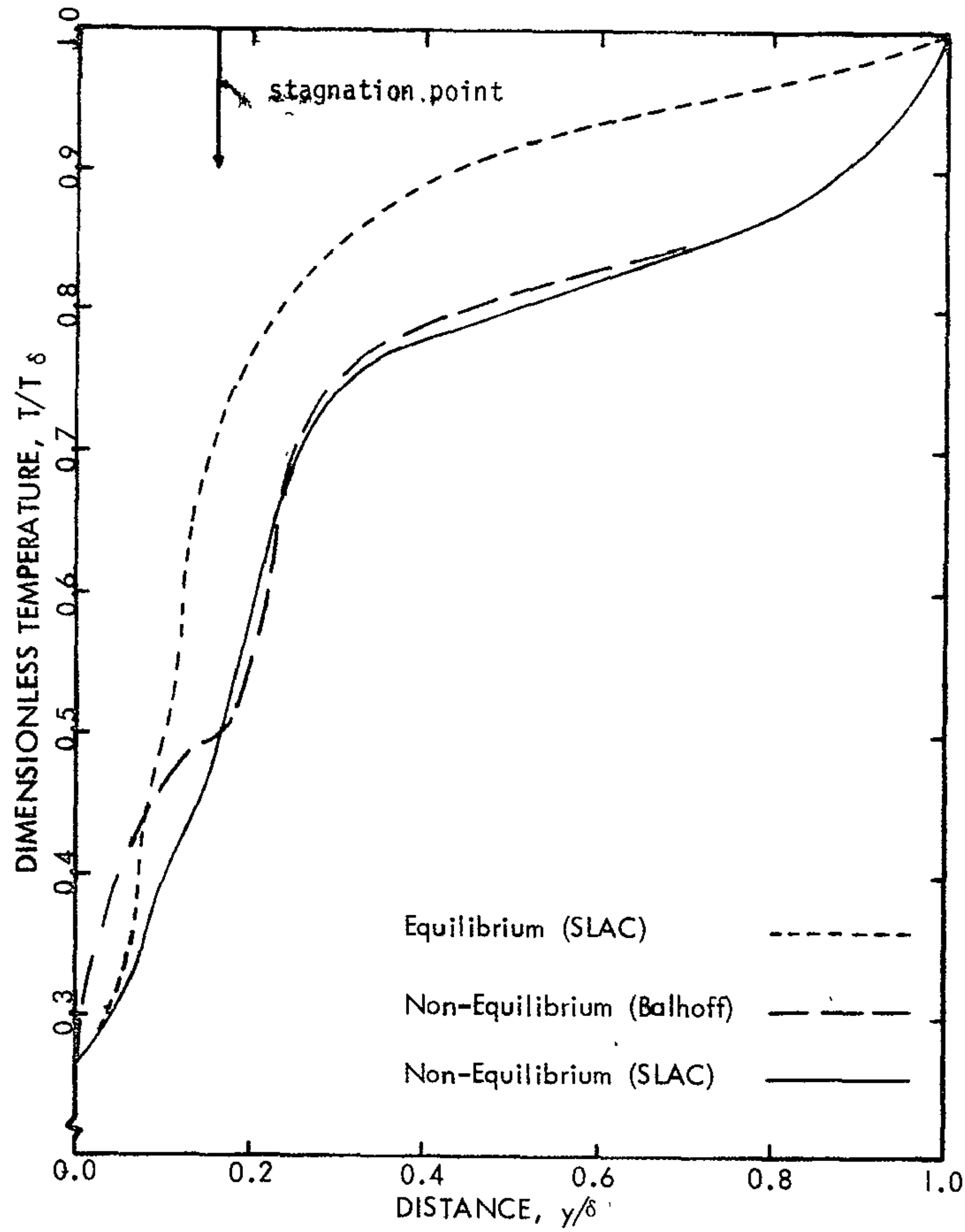


Figure C.1 Temperature Profiles

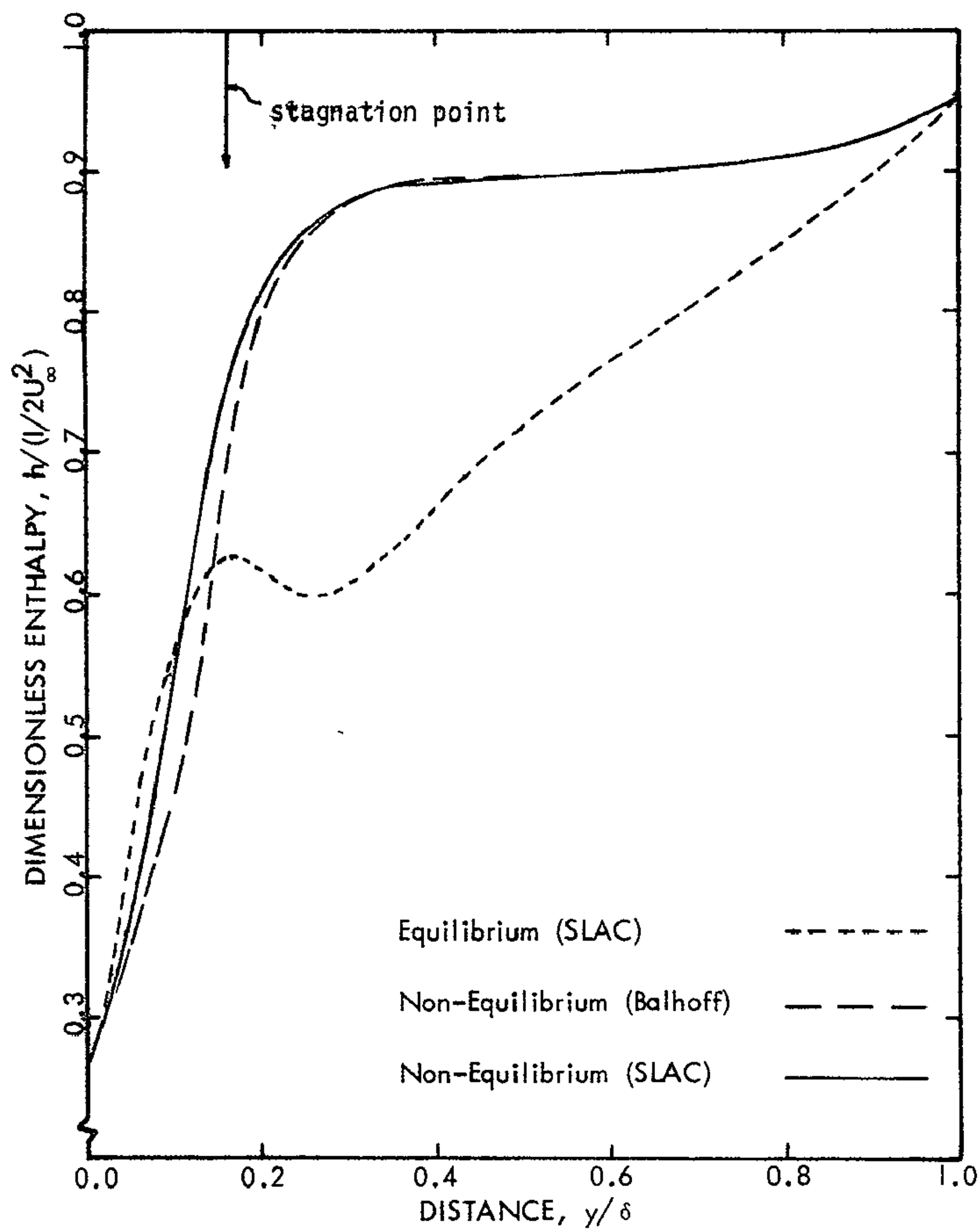


Figure C.2 Enthalpy Profiles

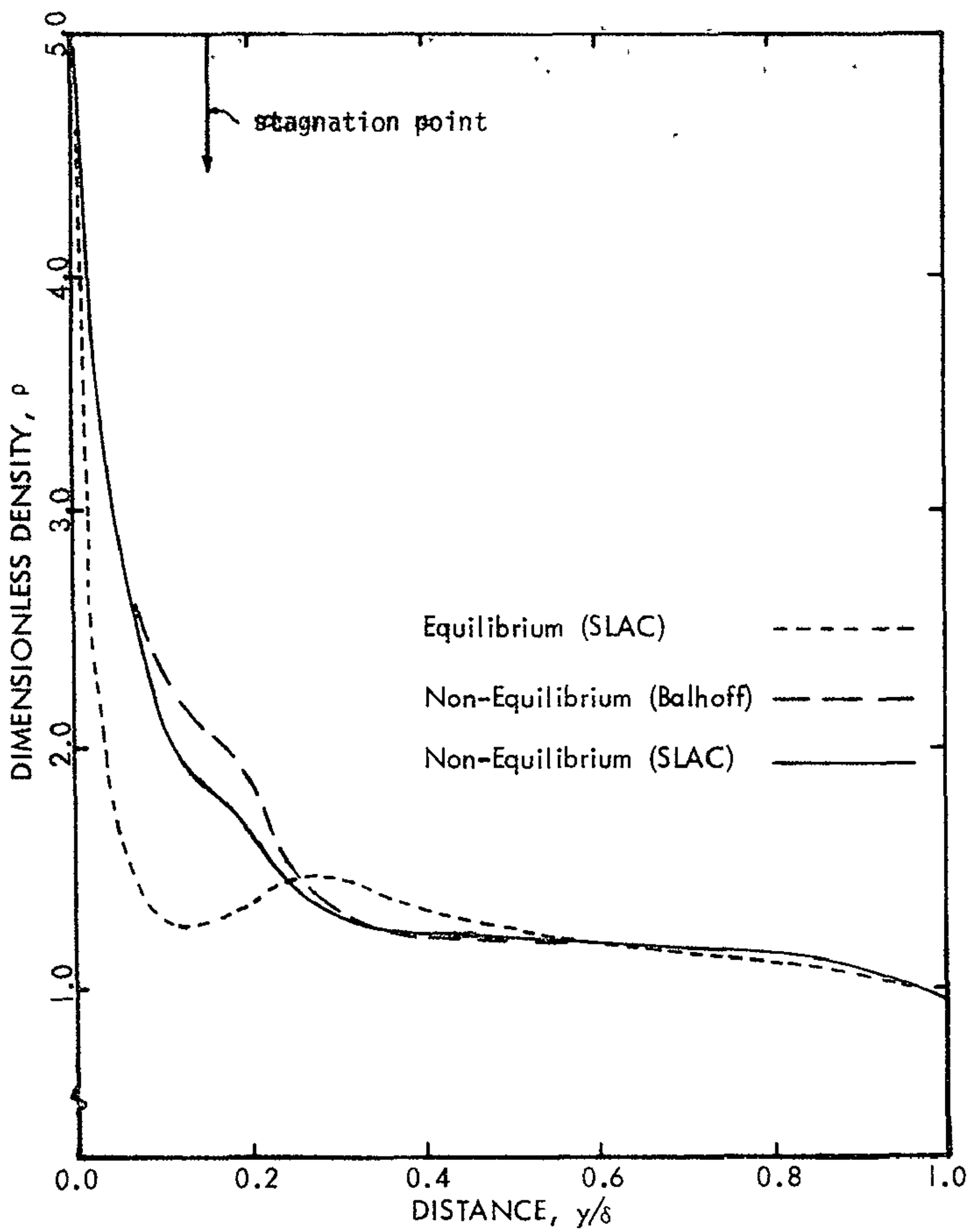


Figure C.3 Density Profiles

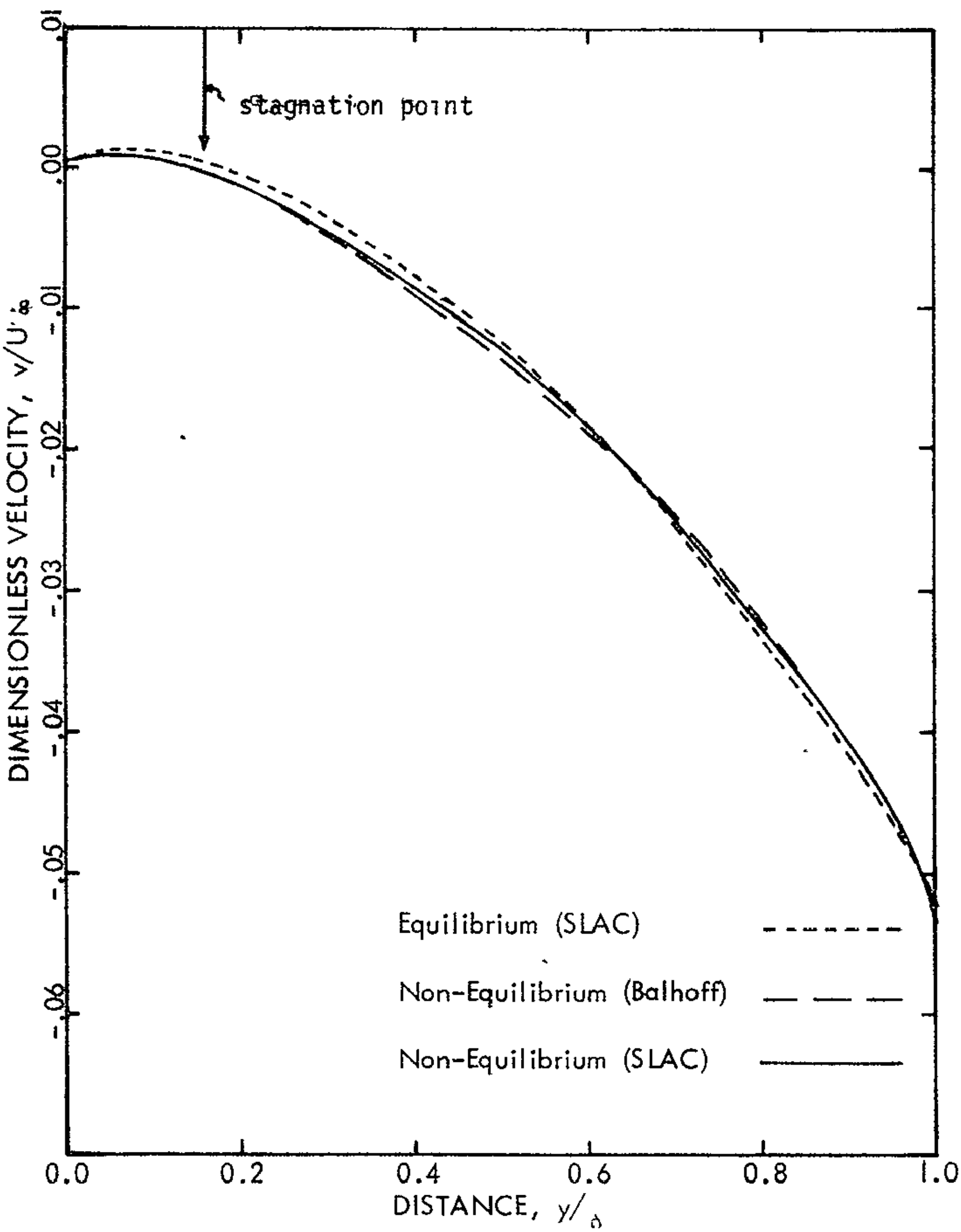


Figure C.4 Velocity Profiles

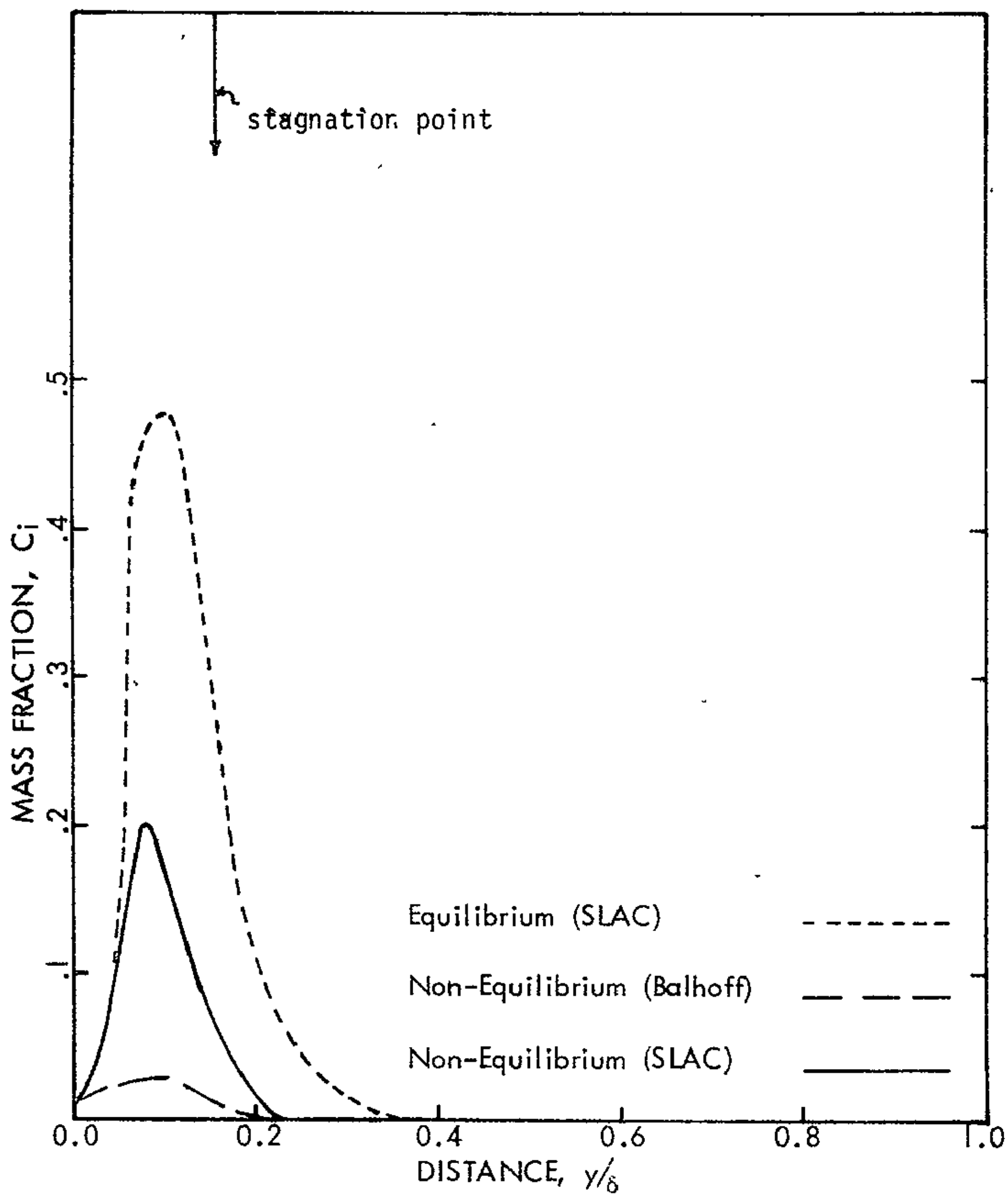


Figure C.5 Mass Fraction Profiles for C

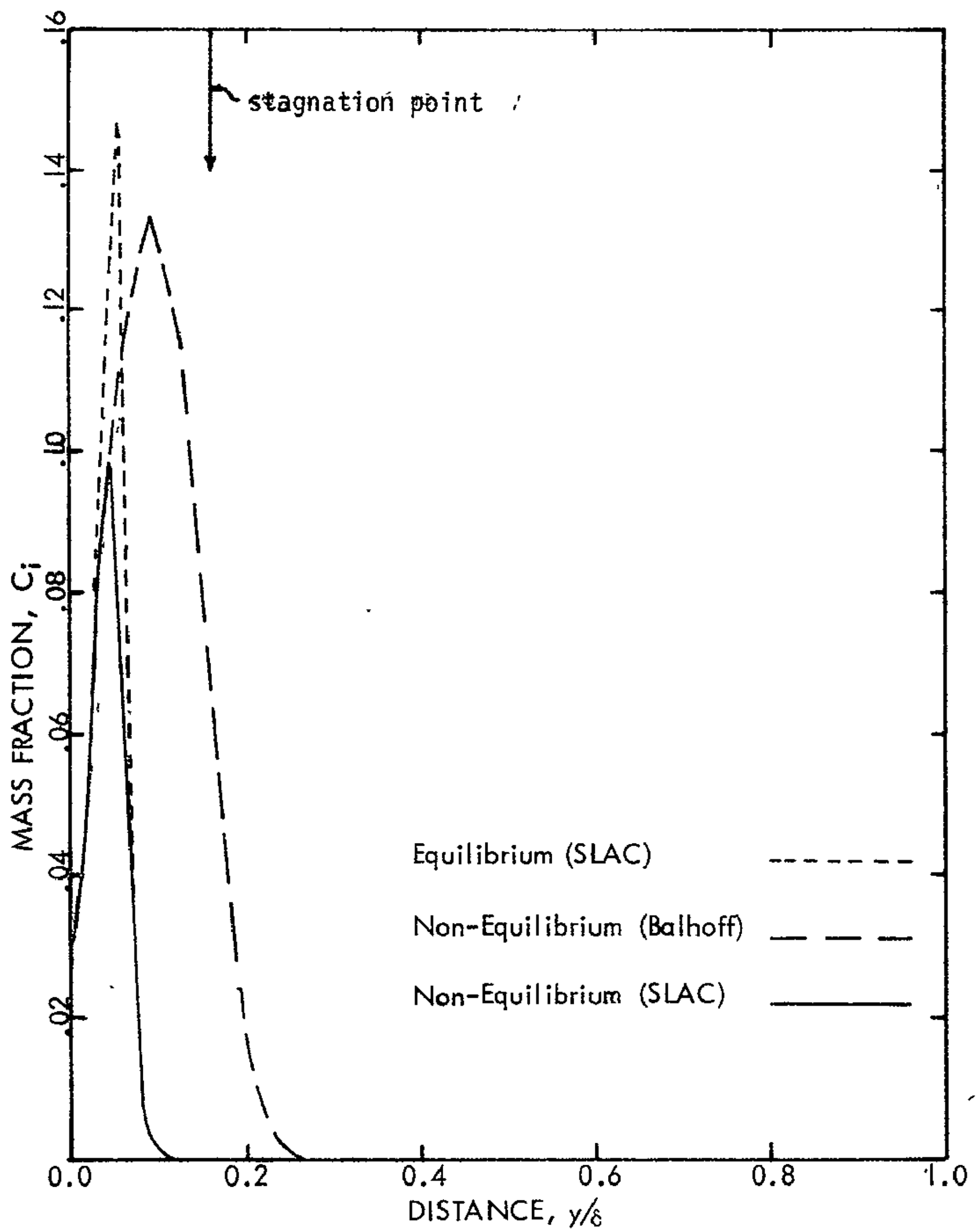


Figure C.6 Mass Fraction Profiles for C_2

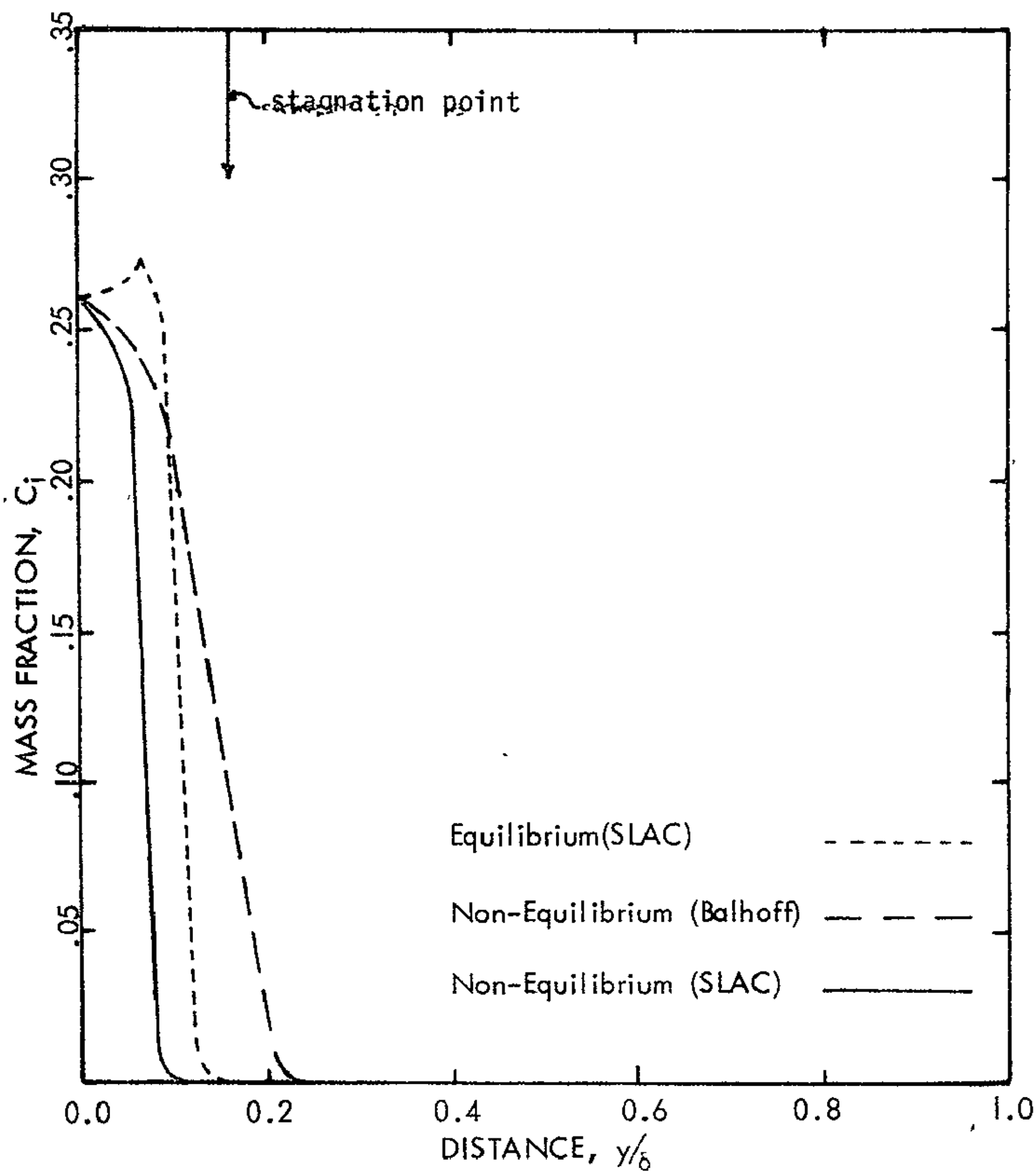


Figure C.7 Mass Fraction Profiles for CO

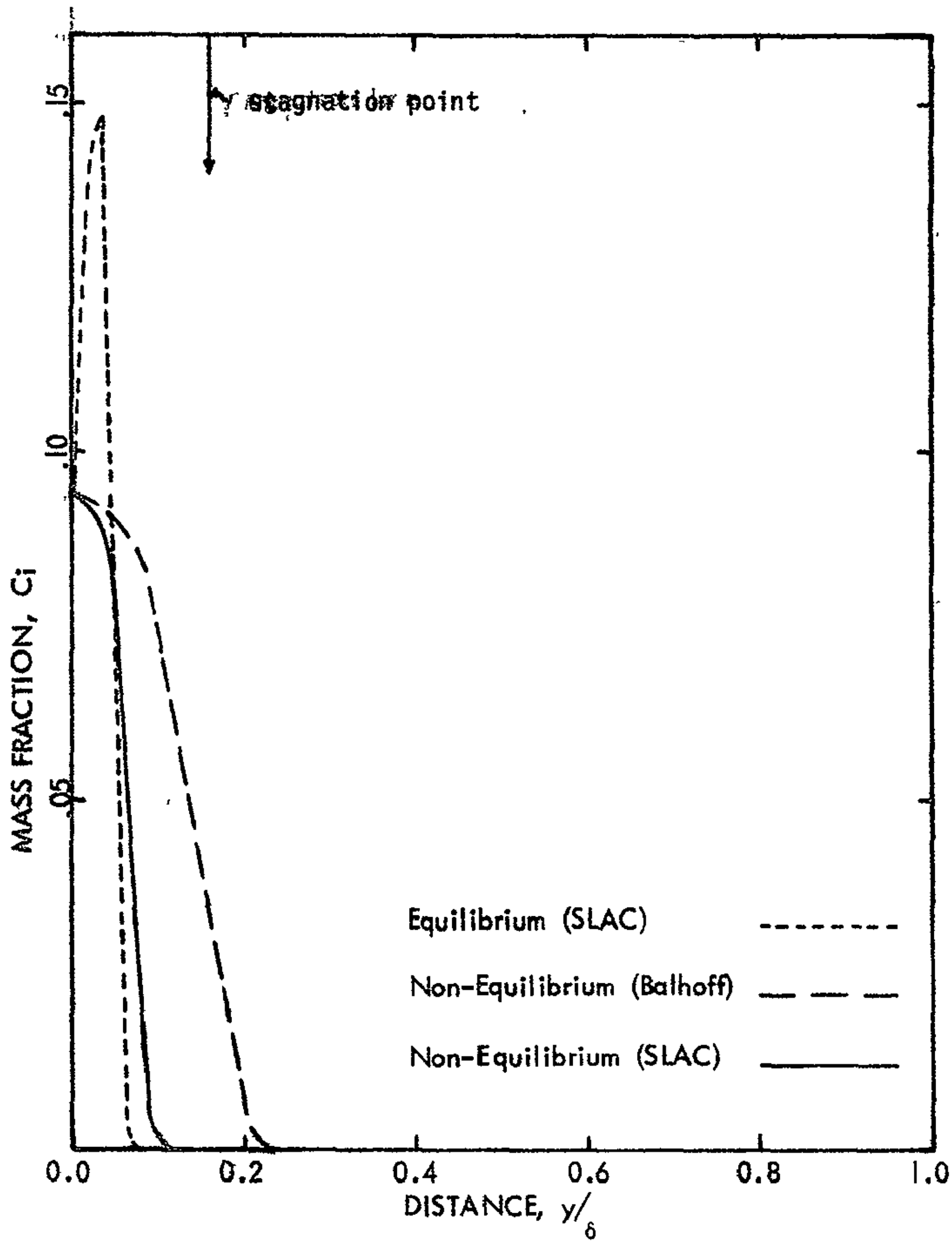


Figure C.8 Mass Fraction Profiles for C_3

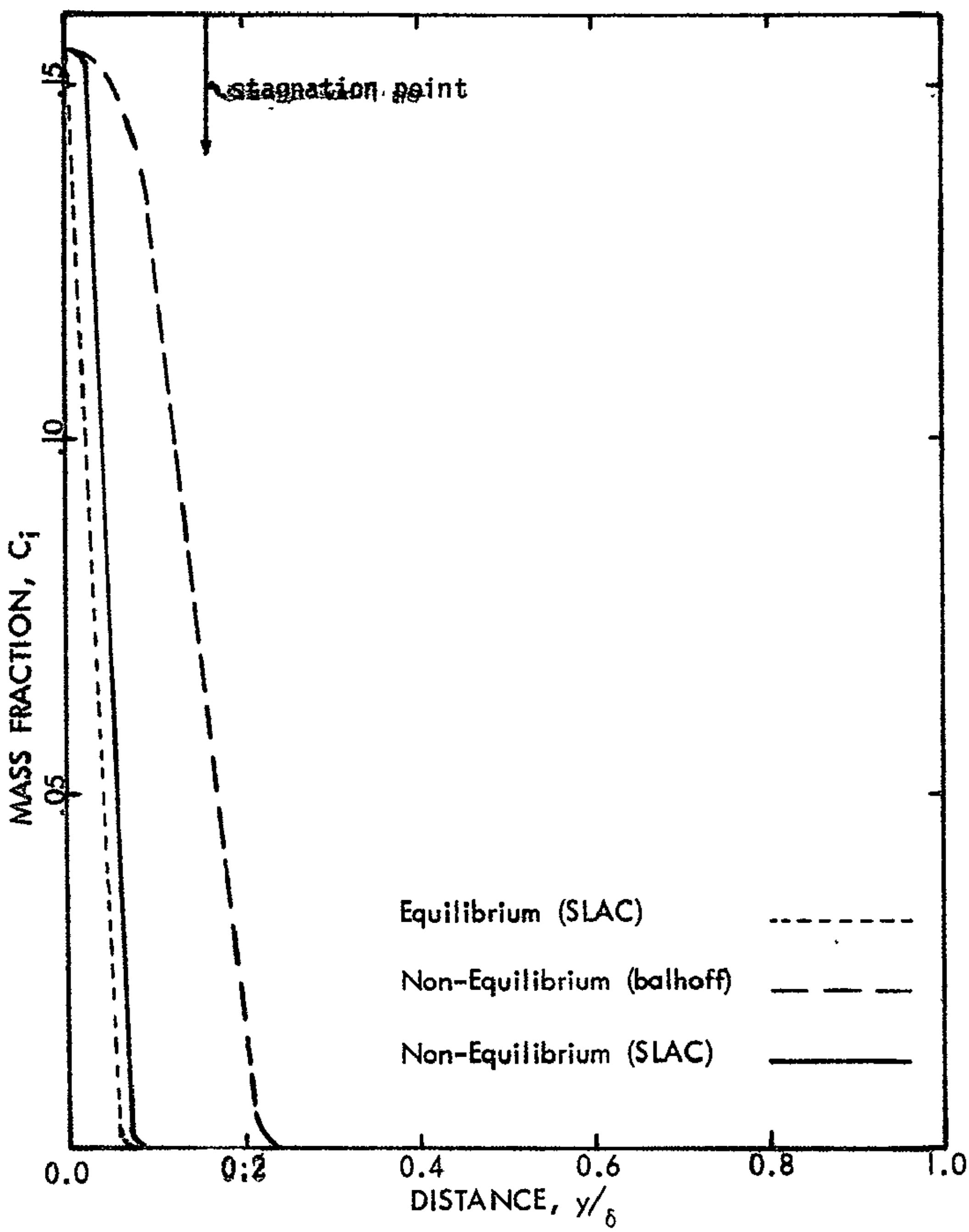


Figure C.9 Mass Fraction Profiles for C_3H

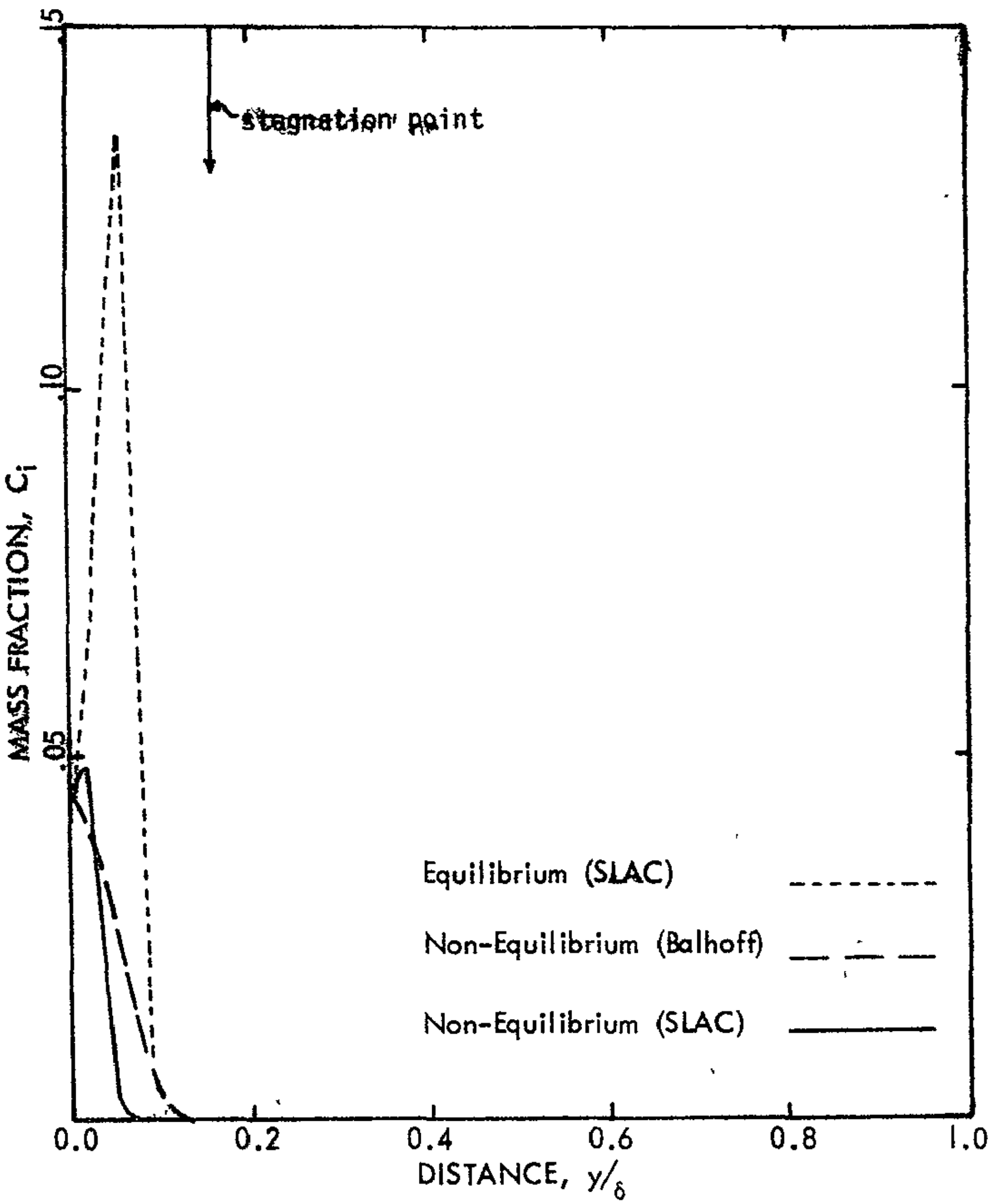


Figure C.10 Mass Fraction Profiles for CN

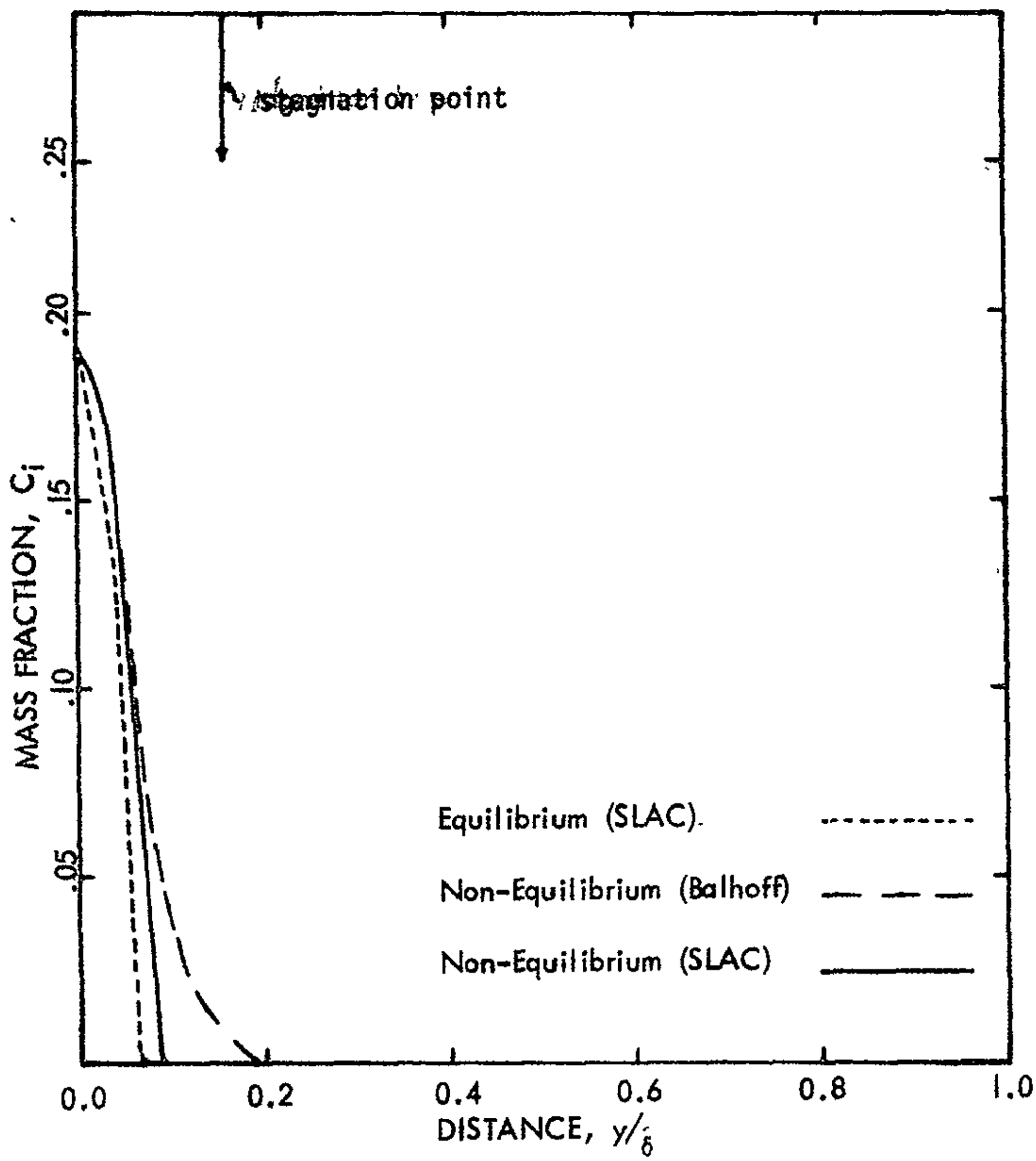


Figure 6.11 Mass Fraction Profiles for C_2H

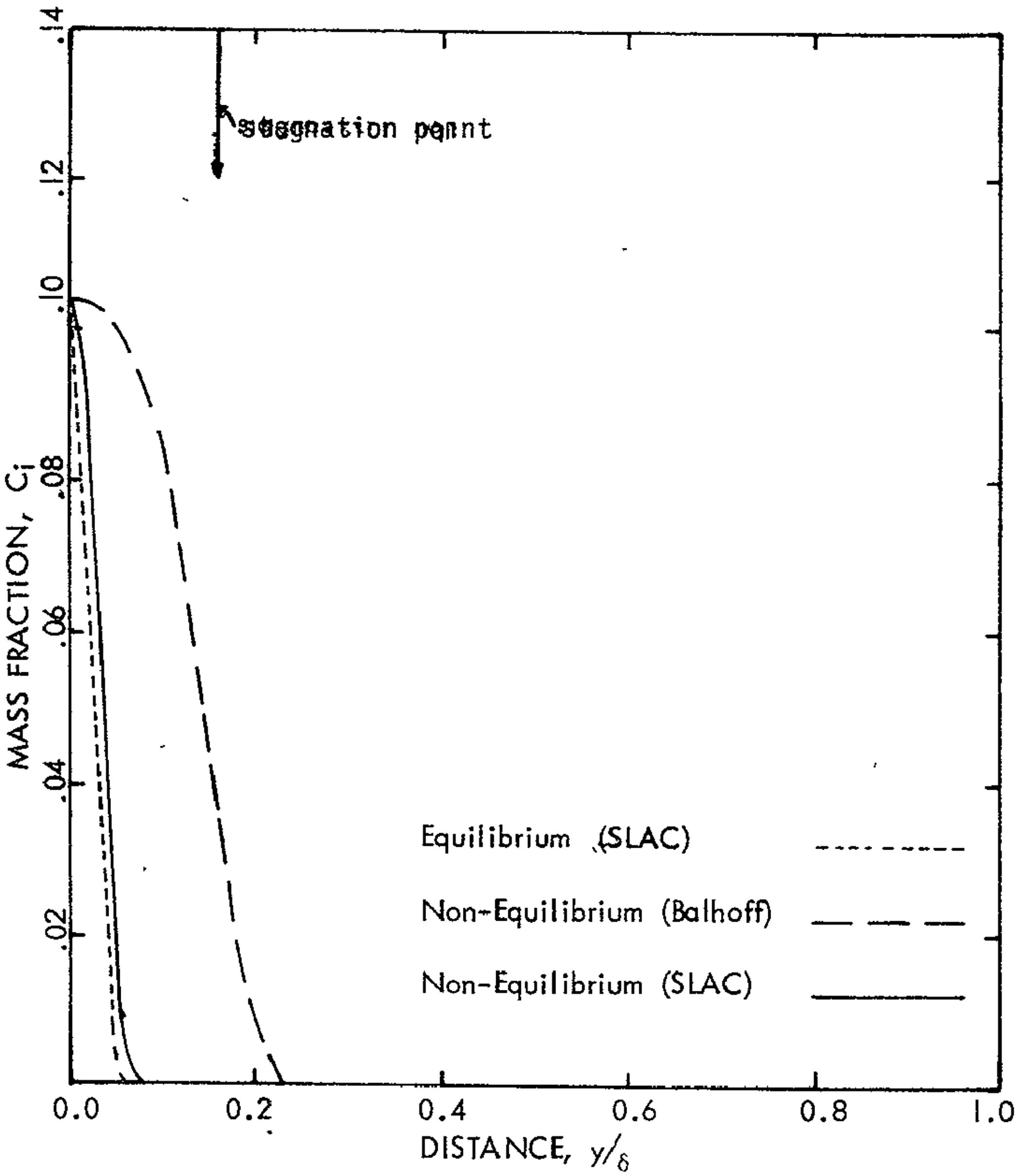


Figure C.12 Mass Fraction Profiles for C_4H

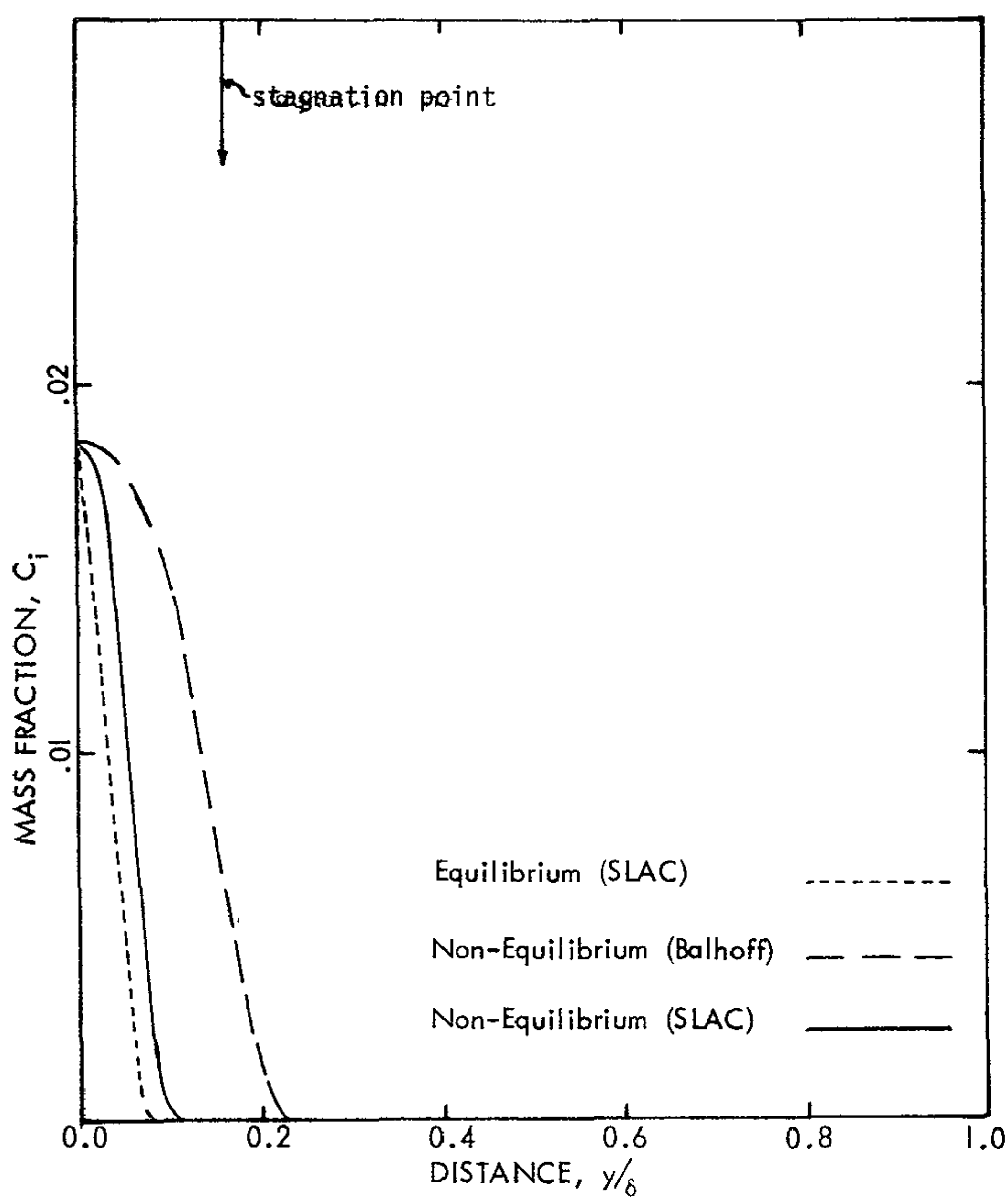


Figure C.13 Mass Fraction Profiles for HCN

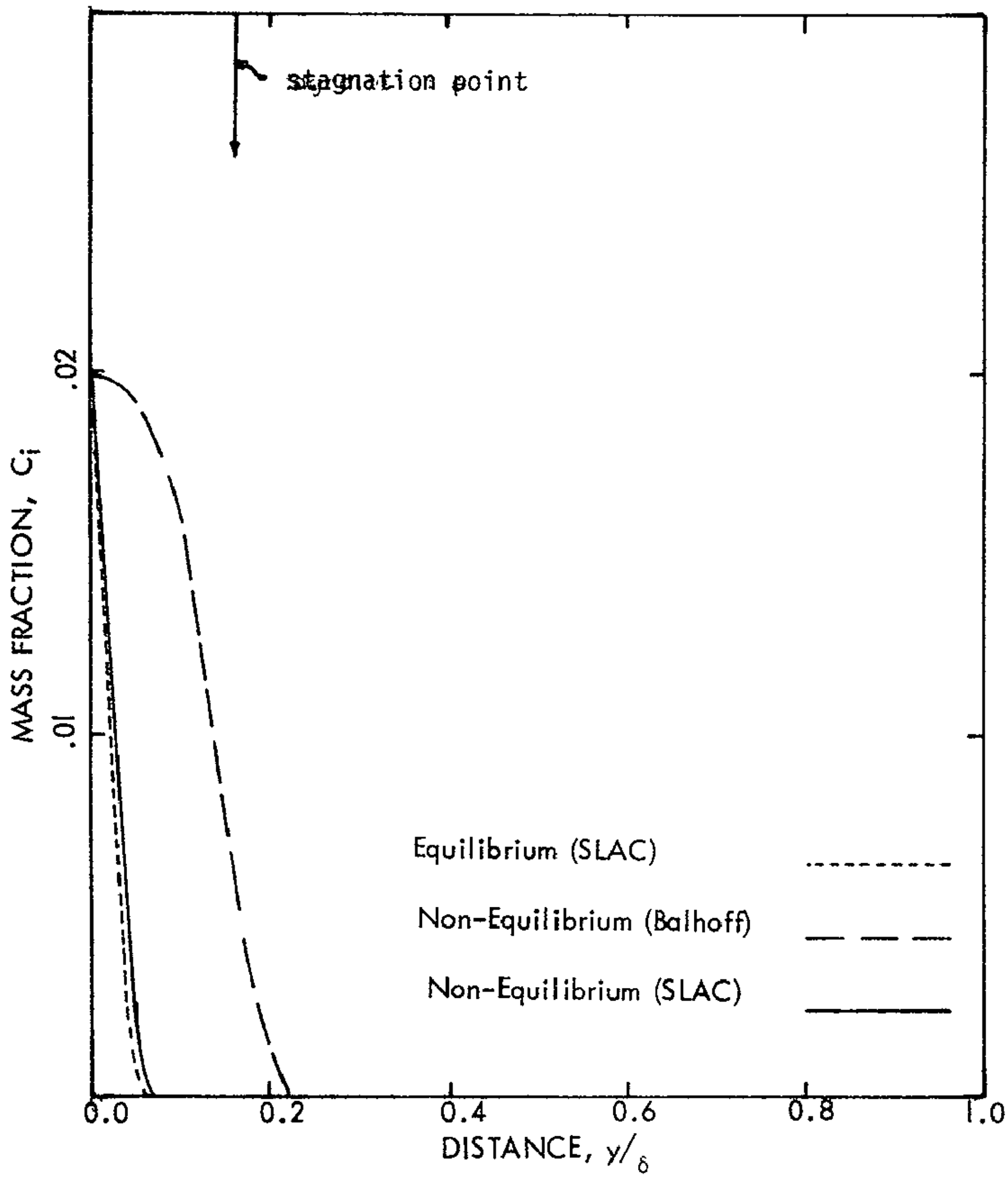


Figure C.14 Mass Fraction Profiles for C_2H_2

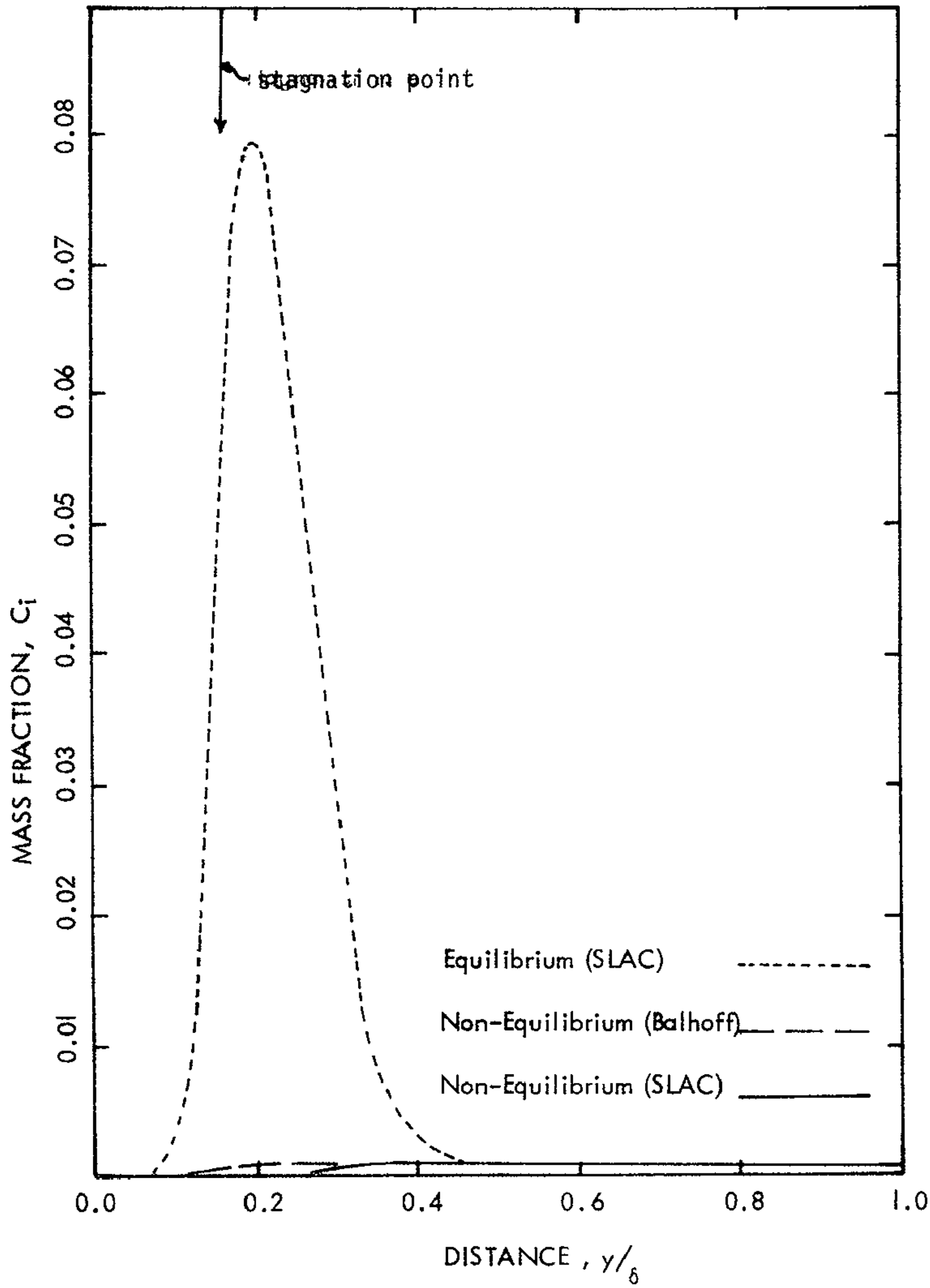


Figure C.15 Mass Fraction Profiles for C^+

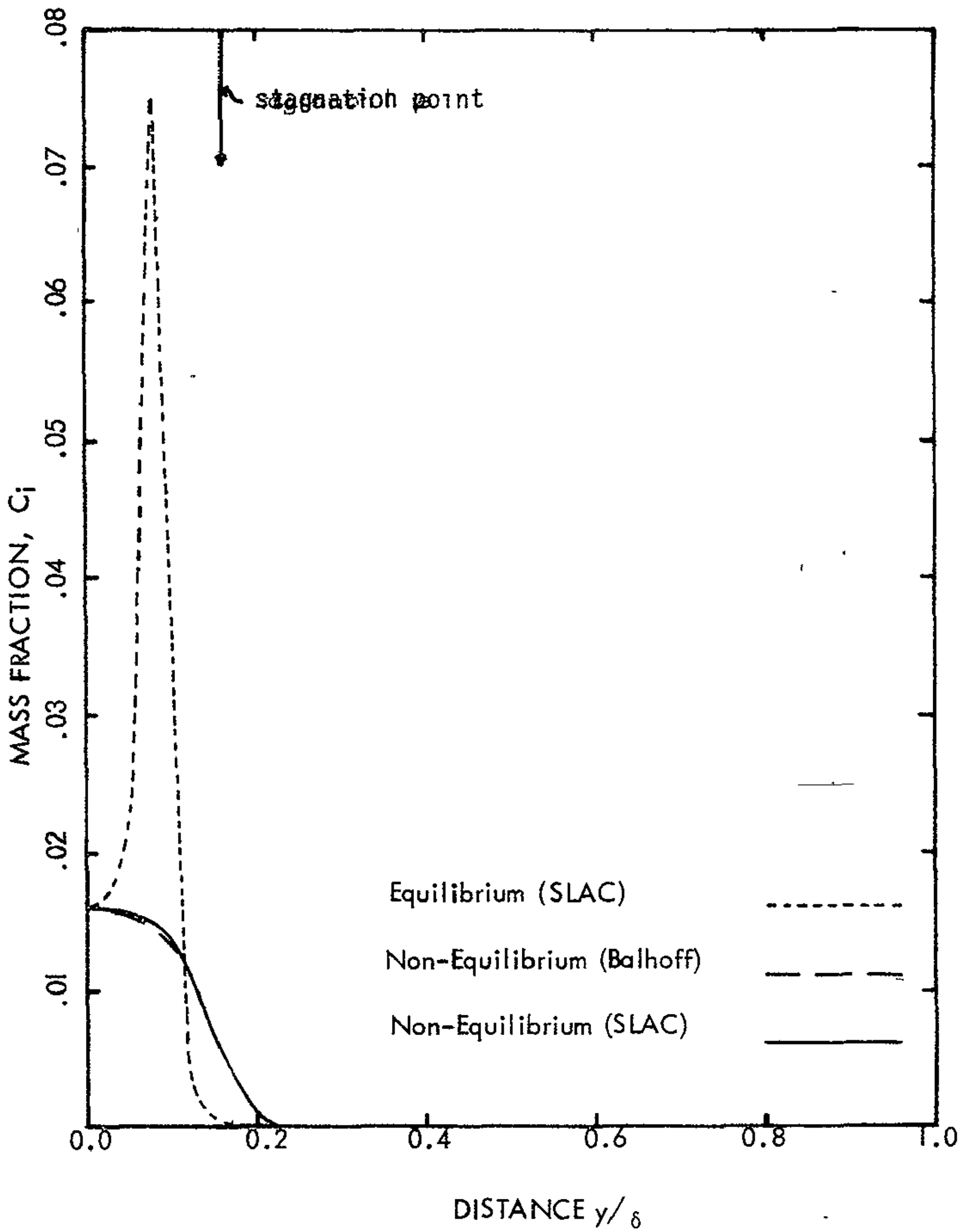


Figure C.16 Mass Fraction Profiles for N_2

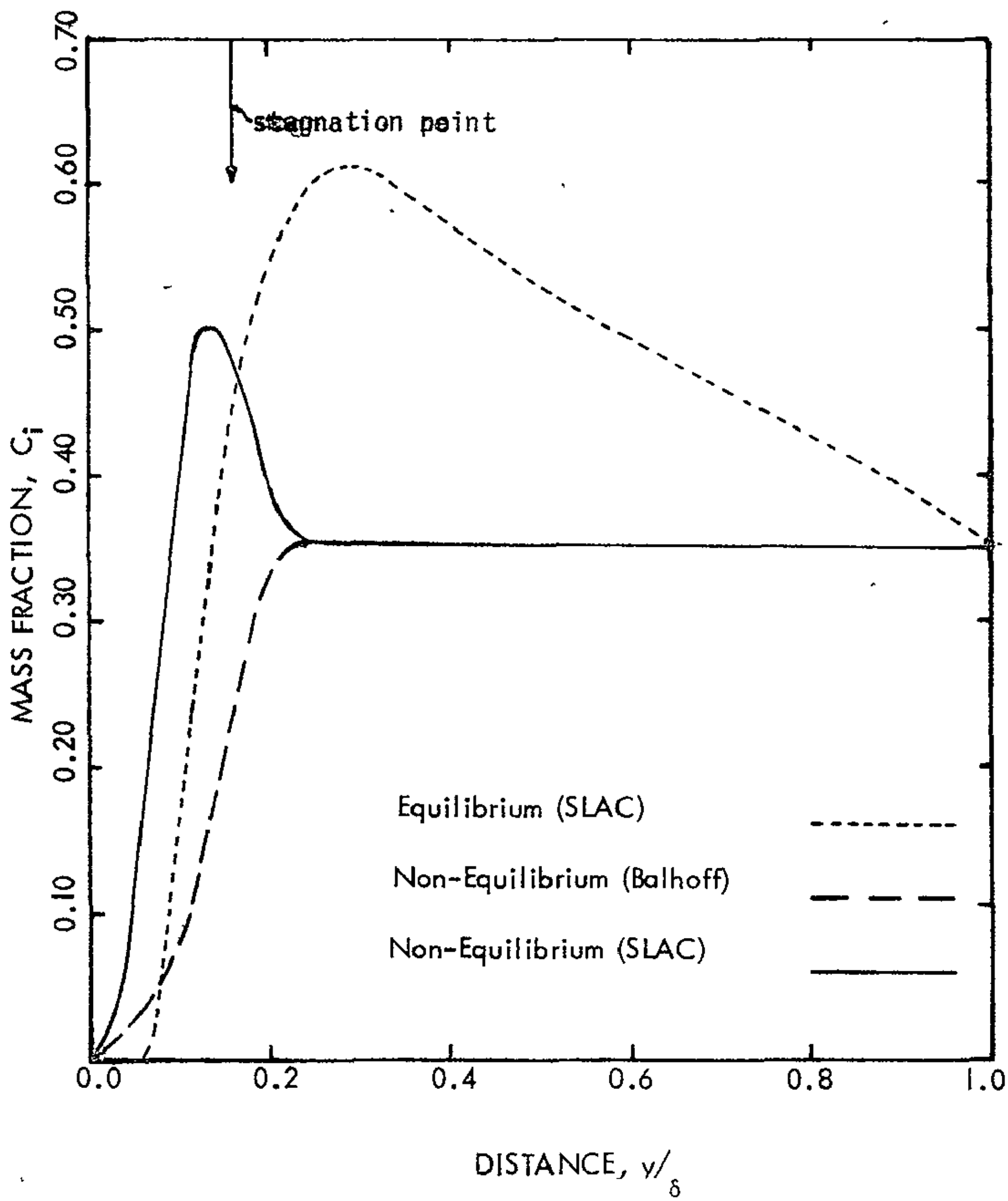


Figure C. 17 Mass Fraction Profiles for N

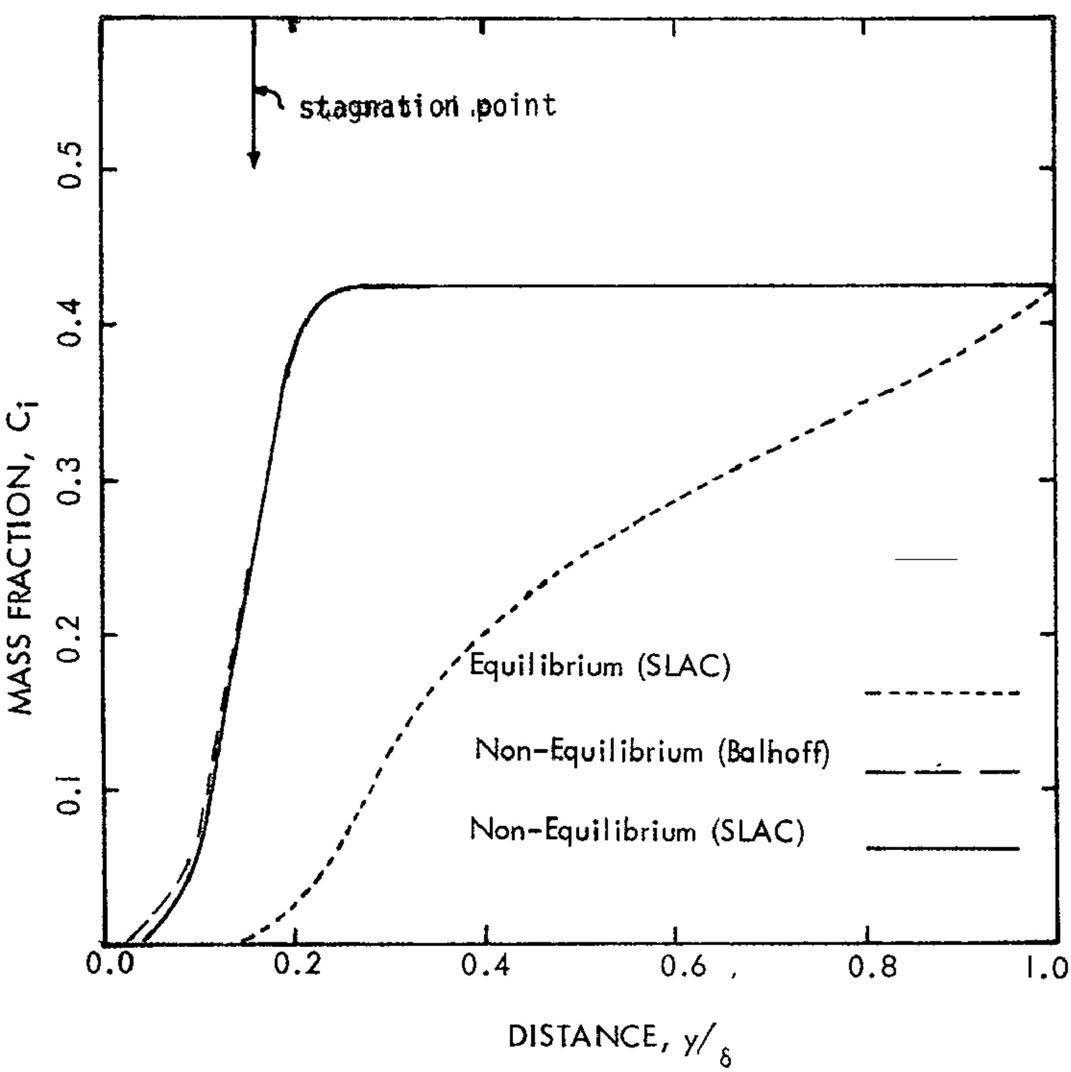


Figure C. 18 Mass Fraction Profiles for N^+

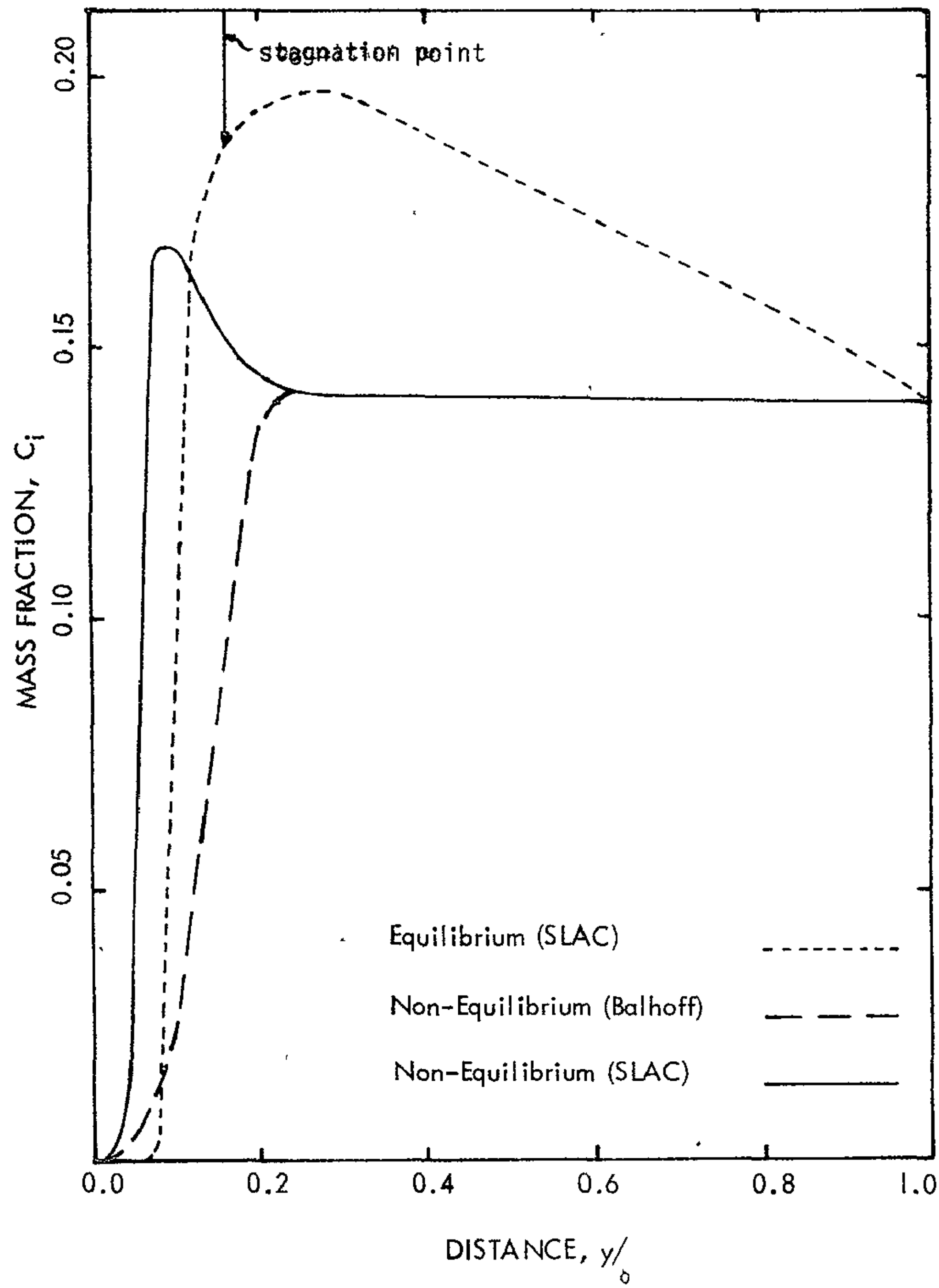


Figure C.19 Mass Fraction Profiles for O

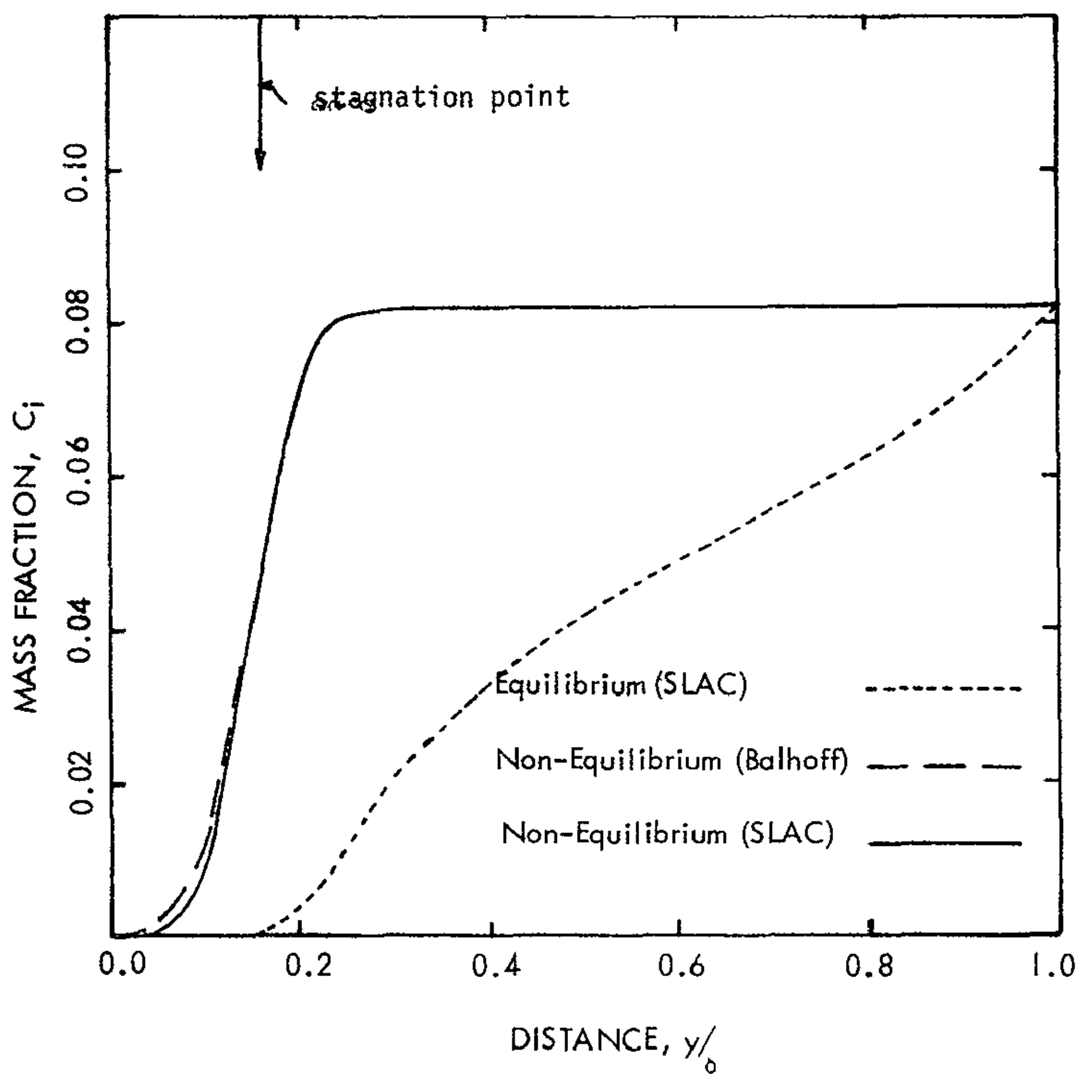


Figure C.20 Mass Fraction Profiles for O^+

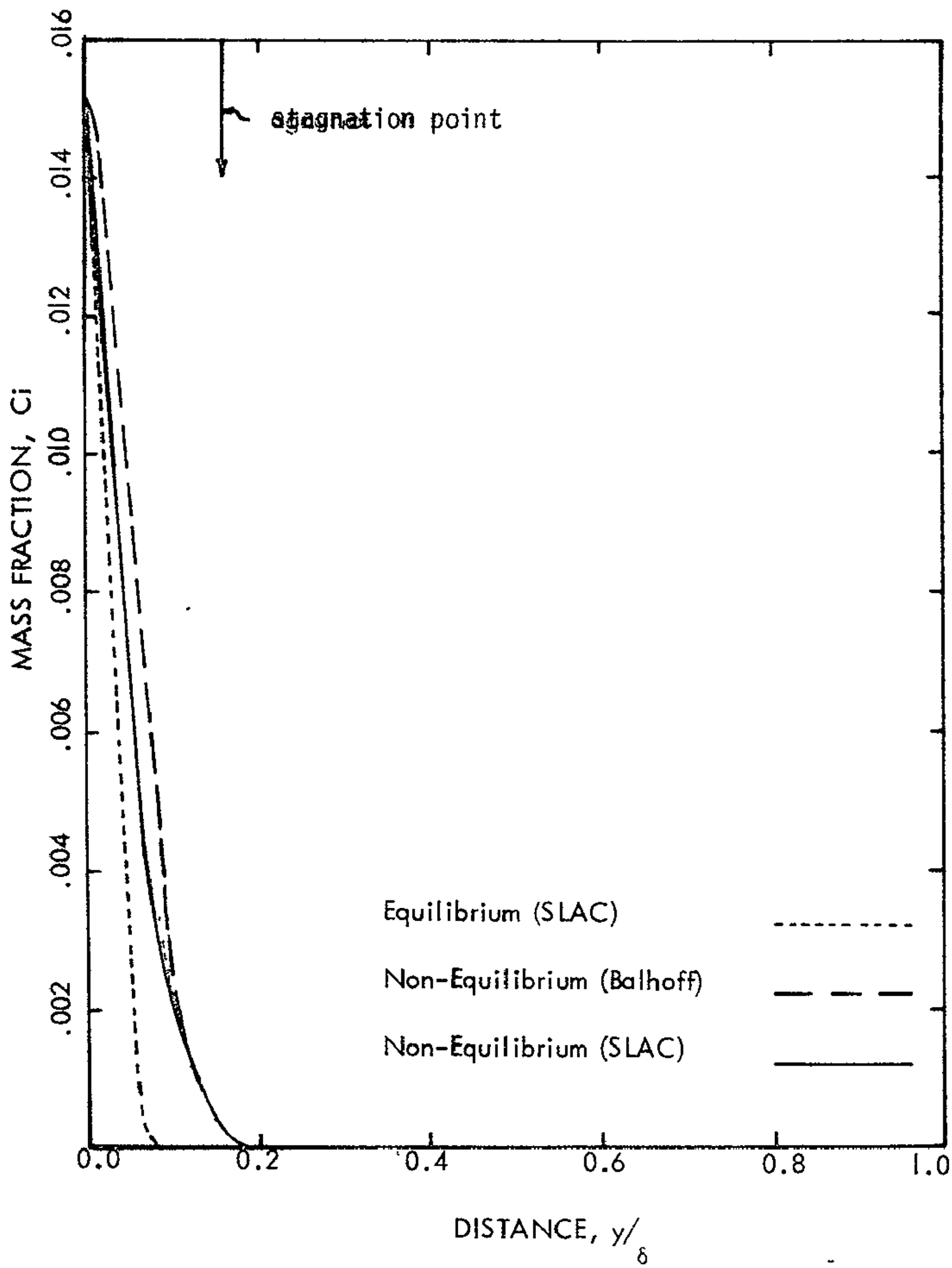


Figure C.21 Mass Fraction Profiles for H_2

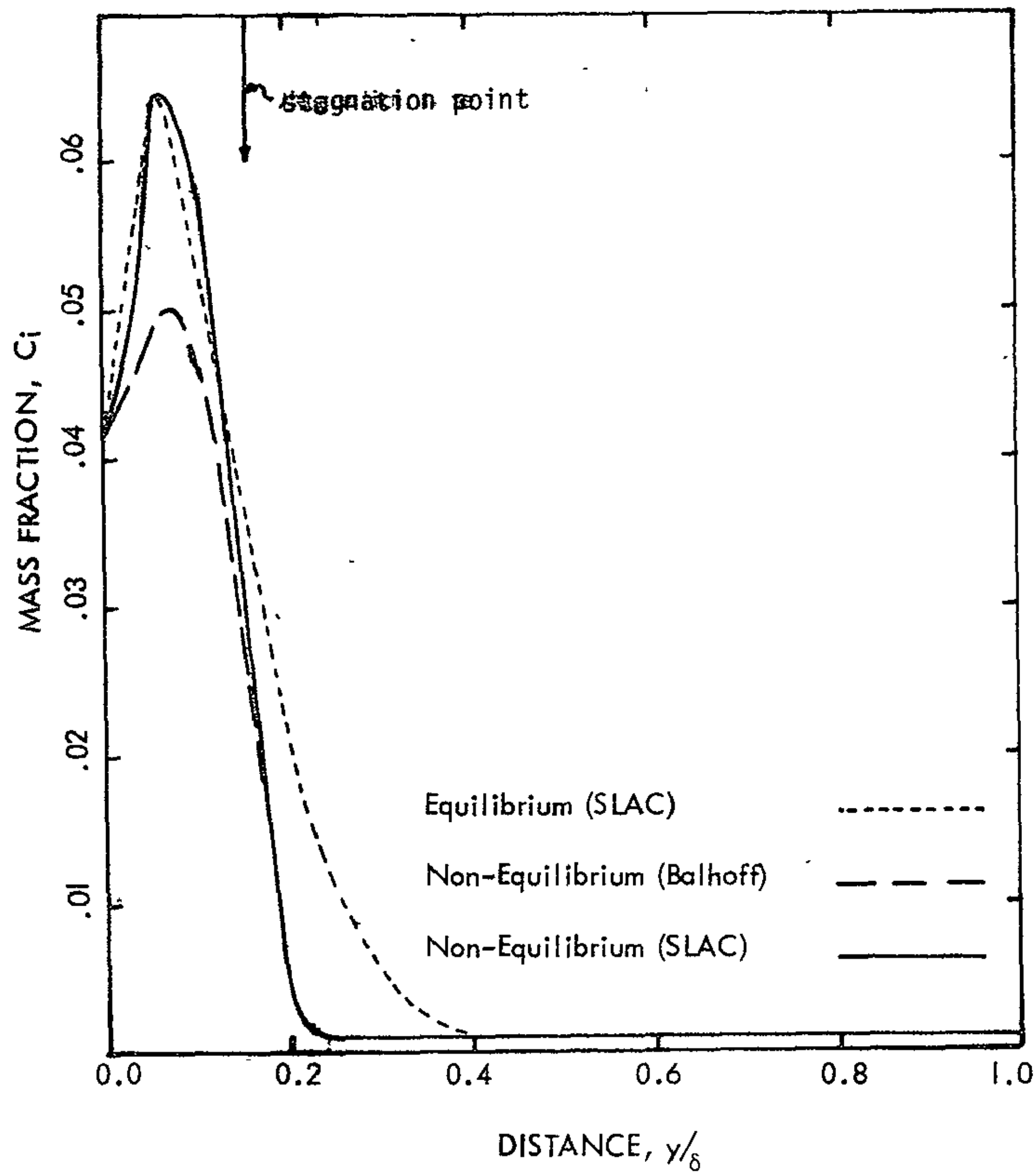


Figure C.22 Mass Fraction Profiles for H

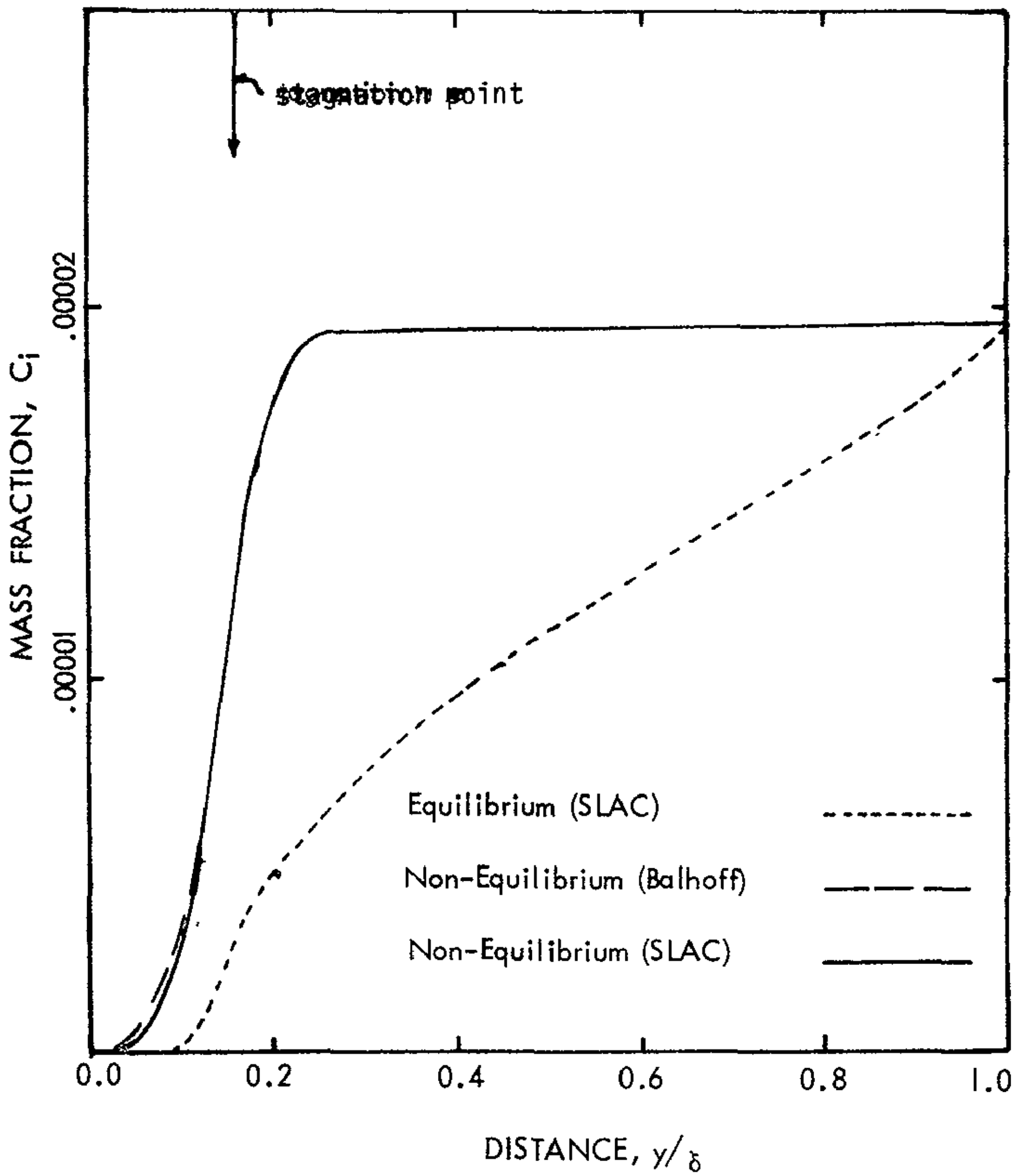


Figure C.23 Mass Fraction Profiles for e^-

VITA

Guillermo Perez was born on [redacted] He completed his secondary education at University High School, Rio Piedras, Puerto Rico in May, 1962. From August, 1962 until May, 1967 he attended the University of Puerto Rico at Mayaguez receiving the Bachelor of Science degree in Mechanical Engineering. In September 1967, following a brief period of employment at the Puerto Rico Water Resources Authority, he enrolled in the Graduate School of Louisiana State University, and in January 1970, he received the Master of Science degree in Mechanical Engineering. From February, 1970 until June 1972 Mr. Perez continued his studies towards the Doctor of Philosophy degree in Chemical Engineering at L.S.U. In July he went to work as Director of the Resource Management Division of Puerto Rico's Environment Quality Board (EQB). While working at the EQB Mr. Perez was subsequently named Director of the Water Quality Bureau (November 1972) and Technical Assistant to the Executive Director (March 1973). Since March 1974, he has been a partner in the consulting firm of Cummings, Perez & Co., a firm specializing in environmental planning. He intends to continue working towards the development of Cummings, Perez & Co.